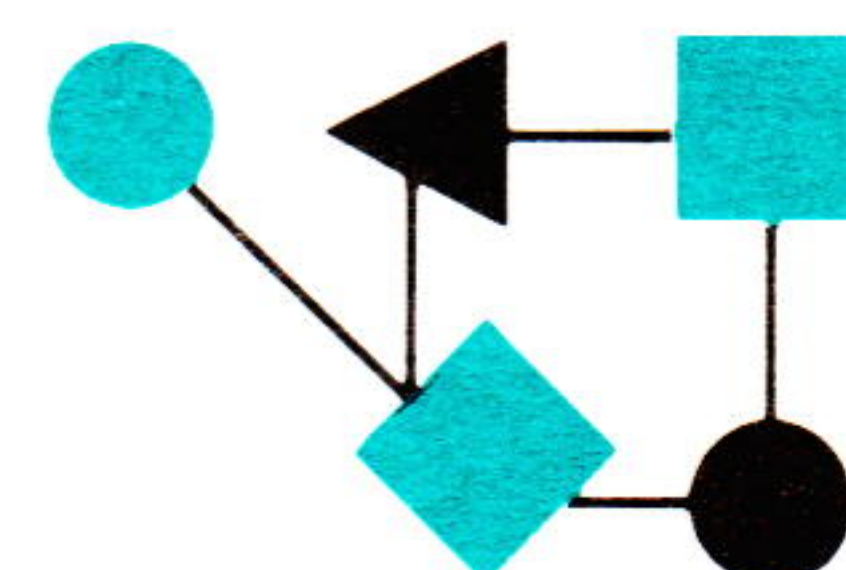


CONNEXIONS



The Interoperability Report

October 1989

Volume 3, No. 10

*ConneXions —
The Interoperability Report
tracks current and emerging
standards and technologies
within the computer and
communications industry.*

In this issue:

The ARPANET is Twenty:

*What We Have Learned
and The Fun We Had..... 2*

Interview with Vint Cerf..... 11

Quotes from some players.... 15

Requiem for the ARPANET.. 27

Book Reviews..... 28

The Telnet Protocol..... 32

*Components of OSI:
CLNP..... 39*

ConneXions is published monthly by
Advanced Computing Environments,
480 San Antonio Road, Suite 100,
Mountain View, California 94040, USA.
Phone: 415-941-3399. Fax: 415-949-1779.

© 1989
Advanced Computing Environments.
Quotation with attribution encouraged.

ConneXions—The Interoperability Report
and the *ConneXions* masthead are
trademarks of Advanced Computing
Environments.

ISSN 0894-5926

From the Editor

Just over 20 years ago, the first nodes of the ARPANET were installed. This marked the beginning of new era. From the ARPANET came TCP/IP, and the advent of true multi-vendor networking. Many have described the ARPANET experience as a "Success-Disaster"—the rapid growth of the Internet has often caused many headaches for the people involved. But the story of the ARPANET is much more than a chronology of technical developments; it is the history of a *community of people* whose lives have been dramatically changed as a result of "the net."

Speaking from a personal perspective, it is certainly true that I would not be writing this column today were it not for a brief visit in 1976 to the Norwegian Defence Research Establishment (NDRE), an early ARPANET site. At NDRE I was introduced to a number of "netfriends"—electronic pen-pals who later inspired me to study this technology, and eventually move to the US.

In this issue, we celebrate the 20th anniversary of the ARPANET with a number of articles and interviews prepared by Daniel Dern. We will focus on the question: "What did we learn from it all?"

This issue is being released at INTEROP™ 89. We hope the conference and exhibition will give you an insight into the state of the art of today's computer networks. Additionally, at the conference, you will be able to see the very first ARPANET node—"IMP 1," on loan from UCLA for this special occasion.

One of the sessions at this conference focuses on *Telnet*, the TCP/IP virtual terminal protocol. In this issue, we have an overview of the Telnet protocol. The article is by Barry Shein of Software Tool & Die. In future issues we plan to explore Telnet further, including a look at the Linemode enhancements which are being developed by one of the working groups of the Internet Engineering Task Force (IETF).

Rob Hagens from the University of Wisconsin describes the Connection-Less Network Protocol (CLNP) in our continuing series *Components of OSI*.

Also in this issue, you will find a number of reviews on recent books in the computer and communications field.



The ARPANET is Twenty: *What We Have Learned and The Fun We Had*

by Daniel P. Dern

Twenty years ago, in 1969, the U.S. *Advanced Research Projects Agency* (ARPA) launched another of its many, serendipitous explorations of interesting technologies.

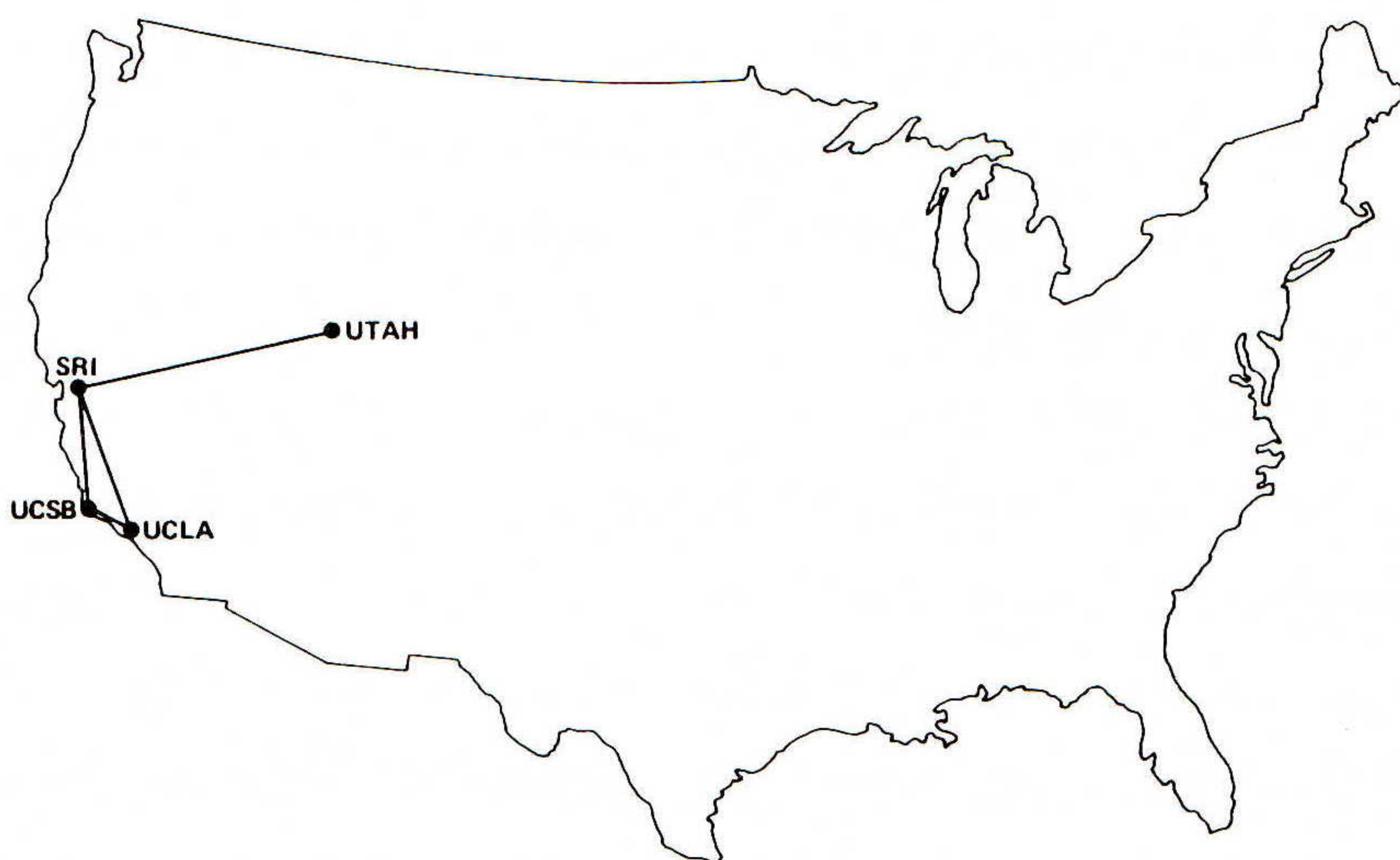
The purpose

This particular project was an experiment in connecting computers—initially, a handful of geographically dispersed, heterogeneous computers, and their users—over a shared network, using a then-new technology. The purpose of this network: to develop techniques and get experience in connecting computers in a way that permitted a very broad range of interactions, such as providing users with remote login access to distant computers, directly or through their own host; permitting sharing of files and other resources; and allowing the use of inter-site electronic mail.

IMP 1 at UCLA

One might argue the year of origin—seminal papers on networking date back to the beginning of the 1960s. But it was January 2, 1969 that teams of software engineers began work, under a contract from ARPA's Information Processing Techniques Office. And it was on September 1, 1969, that the Honeywell 316 minicomputer which was the first of the network's initial four *Interface Message Processors*, or IMPs, arrived at the UCLA campus, air-shipped from Bolt Beranek and Newman (BBN) in Cambridge.

Mirabile dictu, on September 2, 1969, IMP Number 1 started up its software where it left off, three thousand miles ago, and started passing bits to its fellow IMPs at SRI, UCSB, and Utah. It was the beginning of an era. It was the starting point for thousands of projects and careers over the next two decades. It was the birth of an entire industry. It was the start of a new way to think about computers, of a new way to work. It was the start of a state of mind. The name of the technology was packet switching. The name of the network was the *ARPANET*.



The beginning, 1969

This article, and its accompanying interviews and quotes, offers a brief Cook's Tour through the history of the ARPANET—its origins, the twists and turns of its career, and its “children.” In the accompanying interviews and shorter quotes, representative founding and key members of the ARPANET and Internet communities (plus a representative user or two) respond to the questions:

- What was your role in the ARPANET?
- What has the ARPANET meant to you?
- What have we learned from the ARPANET experiment?
- Do any particular incidents or anecdotes come to mind?

What's here is obviously far from everything there is to know about the ARPANET. This is just, if you'll pardon the expression, a few program notes, to our friend, the ARPANET, on the celebration of twenty years of packet switching. (See “Written On The Net,” p. 9).

Birth of a Network

Where did the ARPANET come from? As detailed in the BBN Report 4799, “A History of the ARPANET: The First Decade” (also called the “Completion Report”), the ARPANET traces its primary roots to papers and studies across the decade prior to the development of the first IMP software.

During the early 1960s, Paul Baron and others at the RAND Corporation performed studies whose concepts were to prove central to the ARPANET and its offspring. Ideas in these studies included the superior reliability and survivability of distributed network mesh topologies versus star topologies. They also looked at a form of multiplexing for ‘message blocks’ with headers and data—i.e., *packets*. (See “Packet Switching in 200 Words or Less,” p. 10).

At the same time, Leonard Kleinrock was writing his dissertation at MIT on computer nets, which laid down the foundations for performance evaluation and design of networks, and fellow classmate Larry Roberts was experimenting with networking concepts at MIT's Lincoln Labs. The notion of linking computers came from J.C.R. Licklider, who went on to become the first director of DARPA IPTO. In 1965, the MIT Lincoln Labs commissioned Computer Corporation of America in Cambridge to study the concept of *computer networking*. The primary contact at Lincoln Labs was Lawrence Roberts, who went on to become head of DARPA/ISTO.

By October 1967, discussion towards the proposed ARPA Computer Network, or ARPANET for short, was going like gangbusters. Topics included message formatting, message protocols, dynamic routing and message propagation, queuing, error control, and node-to-host communications. In July 1968, ARPA mailed out an RFQ for the network. From the twelve responses, a contract was awarded in mid-December 1968 to Bolt Beranek and Newman of Cambridge, Mass. Under the leadership of Frank Heart, a group set out to develop software for the first IMP.

On January 2, 1969, work began. And on September 1, 1969, the net was born.

Networking for success

The ARPANET succeeded beyond anyone's wildest dreams—anyone except perhaps its founders, like Larry Roberts, Bob Kahn, and Vint Cerf, who could see the future lurking implicit in their off-spring's electronic insides.

continued on next page

The ARPANET is Twenty (*continued*)

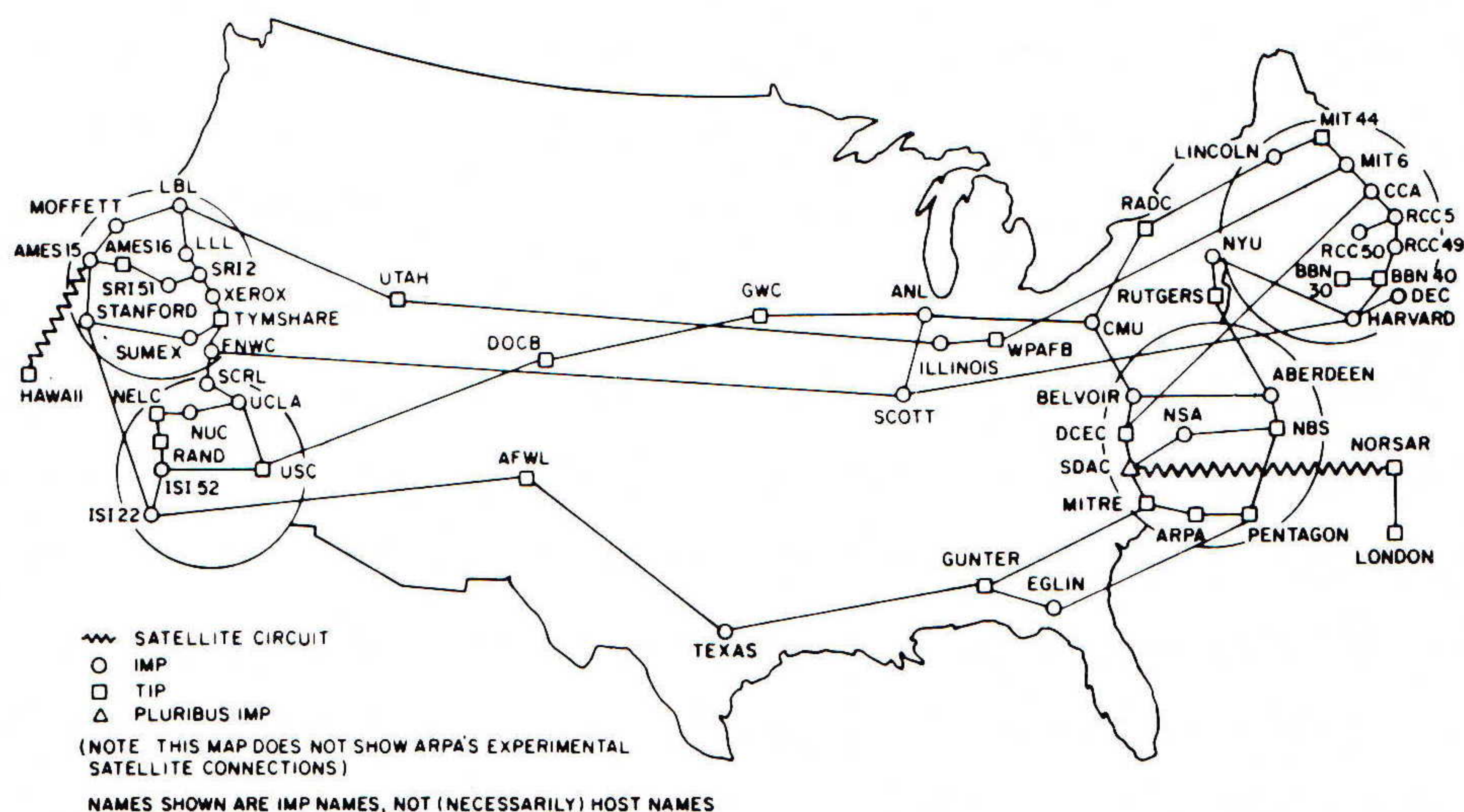
Growth

The new network grew—slowly at first, then more rapidly. And it kept growing. For over a decade, the ARPANET grew by an average of one new host computer every 20 days.

At its peak, before the ARPANET/MILNET split in 1983, there were over 100 packet switches. Increasingly, the ARPANET was not only used as a testbed for network technology, but also as the agent for “operational” traffic—i.e., to carry messages as well as experiment in whether the network could carry them.

TCP/IP

By 1975, the network was functionally an operational Department of Defense (DoD) network. Control was transferred from ARPA, which had become DARPA by this time, to DoD, under the U.S. Defense Communications Agency. Here, ARPANET technology became the basis of the Defense Data Network program, the DDN, a multi-level, multi-network program intended to serve the world-wide datacomm needs for the DoD. (See *Quotes: Heidi Heiden*, p. 16). NCP, the original Network Control Program, was replaced by the more robust *Transmission Control Protocol/Internet Protocol*, TCP/IP.



December 1976

For an experimental program, the ARPANET was inarguably a rousing success. According to the BBN Completion Report on the network's first decade, “The success of the program far exceeded even the most optimistic views at the time of inception.” Technical successes stemming from the ARPANET programs—many of which were not on the original ‘shopping list’—included numerous landmark demonstrations:

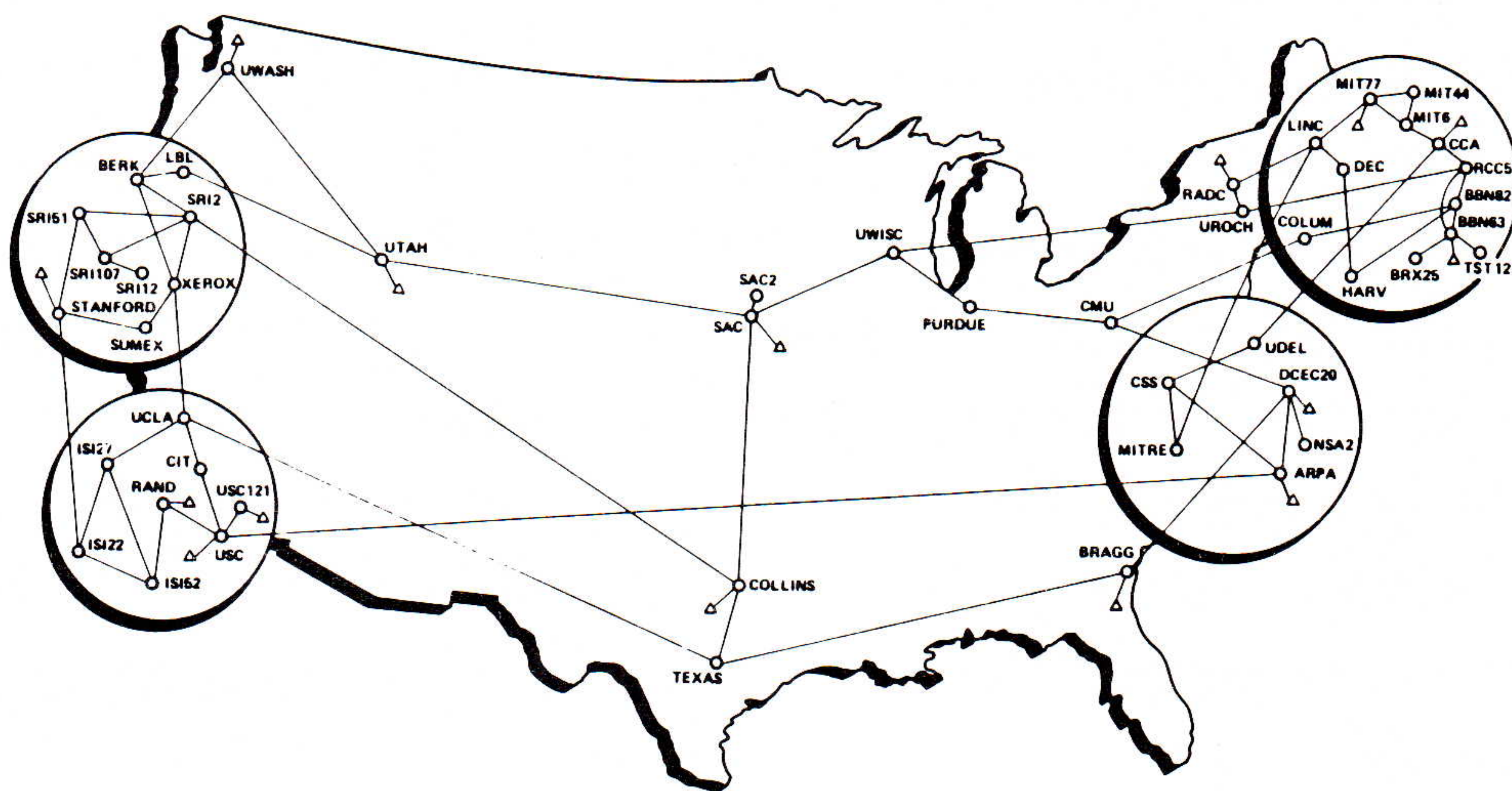
- That a packet network with adequate performance, reliability and cost promoted resource sharing
- That adaptive routing algorithms could provide reliability, efficiency, and flexibility in service
- That a large operational network could be built where local failures would have minimal impact on overall service

- That a network can be constructed in a way that nodes, lines and capacity can be added or deleted without major interference in operations
- That a network can control and operate itself for long periods (hours to days) without explicit human control from an operations center
- That a community of different computers and operating systems could communicate—perhaps the most difficult task undertaken in the development of the ARPANET

And we learned many, many other lessons—the value of e-mail, the importance of reliability, how to work together. Founding figures like Robert Kahn, Lawrence Roberts, Vint Cerf, and others make this clear, as do more recent net mavens like Eugene Spafford and Geoff Goodfellow, in the accompanying interviews and quotes.

The Split

By the early 1980s, it became clear that the ARPANET was not merely carrying operational traffic for the R&D communities, but also for an increasing portion of military sites. In 1983, the ARPANET was split, first logically and then physically as well, into two separate networks; the MILNET, DDN's unclassified operational military network, and the ARPANET, once again a research testbed and non-military traffic highway. The ARPANET/MILNET *Split*, as this process was called, was notable for the smooth manner in which it was carried out. For the most part, user service continued undisrupted throughout—a tribute both to careful planning and the highly adaptable nature of the protocols running the net.



March 1986

Hackers, Crackers, Snoops and Spies

No article about the ARPANET would be complete without a few good war stories. Many are part of the oral (and electronic) history—the great FINGER controversy, Black Tuesday, “Gateway Wars” to name a few. But there are two that have made it beyond net folklore, into the world at large...

For the most part, the life and times of the ARPANET have stayed within the confines of the various electronic mailing lists, technical and trade journals, and symposia. But every so often, something happens which catches the public eye. For example, the case of the extra account...

continued on next page

The ARPANET is Twenty (*continued*)

Until August 1986, astronomer Cliff Stoll was, in his own words, “just another user on the net.” Cliff was using the ARPANET for his work in astronomy, was sending technical specifications and data back and forth. His responsibilities at UC Berkeley included system administration. And one day that August, he noticed an accounting error in one of their computers.

“We discovered somebody had illegally added an account,” Stoll recalls. “Instead of locking them out, we cleverly and quietly watched them. We found they were breaking into our computer, and from there systematically breaking into other ARPANET and Internet sites, into military systems all over the country. Over the next year, we watched the intruders, tracked them down—and finally caught them. Now I’m also a computer security expert.”

Stoll’s adventures made the national news—*The New York Times*, *The Wall Street Journal*, and others. Stoll himself documented his network spy-tracing in his article, “Stalking the Wiley Hacker,” which appeared in the May 1988 issue of *Communications of the ACM*. (A book-length version of his adventures, *The Cuckoo’s Egg*, will be published by Doubleday this fall.) Two years later, something else happened—a more than nine-day wonder which not only put the ARPANET/Internet clearly into the public view, but...

The worm that ate the Internet

“We are currently under attack from an Internet Virus.” That was the beginning of an e-mail message from Peter Yee at NASA Ames Research Center, to the Internet’s TCP-IP mailing list, at 11:28 PM, Pacific Coast Time, Wednesday, November 1, 1988.

“It has hit UC Berkeley, UC San Diego, Lawrence Livermore, Stanford, and NASA Ames. The virus comes in via SMTP, and then is able to attack all 4.3BSD and Sun (3.X?) machines. It copies in a program, compiles and executes it. The program copies in VAX and Sun binaries that try to replicate the virus via connections to *telnetd*, *ftpd*, *fingerd*, *rshd*, and SMTP. The programs also appear to have DES tables in them. They appear in `/usr/tmp` as files that start with the letter x. Removing them is not enough as they will come back in the next wave of attacks. For now turning off the above services seems to be the only help. The virus is able to take advantage of `.rhosts` files and `hosts.equiv`. We are not certain what the final result of the binaries is, hence the warning.”

With this message began a bizarre series of days that felt like a cross between a computer detective novel and the War of the Worlds. A self-replicating worm, unleashed either deliberately or accidentally, ran amuck across the ARPANET and Internet. By the time it was done, thousands of host computers were infected, the connectivity of the Internet itself disrupted—and the world all too aware of networks and worms.

Within hours, the story was on national television and radio news, and on the front page everywhere from the *Boston Globe* to the *New York Times* and *Wall Street Journal*. Within days, it not only reached all the computer trade publications, but even local papers in upstate Vermont, *USA Today*, the *Bloom County* comic strip, and Erma Bombeck’s column.

For weeks and months after, the story has stayed news—the hunt for the alleged virus maker, the aftermath, editorials on “What have we learned and how do we prevent recurrences,” and decreasingly-sized stories on the various studies, committees, findings and whatnot. My own clipping file from the Worm reached about two inches thick, before I stopped. My personal favorite headline, from the November 21, 1988 issue of *MIS Week*:

“Virus Suspect’s Mom Stands By Her Son”

The Internet community itself wrote a number of cogent post-mortems on the Worm. Donn Seeley, from the University of Utah, in the concluding remarks in “A Tour of the Worm,” noted, “Our community has never been in the limelight in this way, and judging from the response, it’s scared us. I won’t offer any fancy platitudes about how the experience is going to change us, but I will say that I think these issues have been ignored for much longer than was safe, and I feel that a better understanding of the crisis just past will help us cope better with the next one. Let’s just hope we’re as lucky next time as we were this time.”

Eugene H. Spafford, at the Department of Computer Sciences at Purdue University in West Lafayette, Indiana, concluded, in his article “The Internet Worm Program: An Analysis,” (*ACM SIG-COMM Computer Communication Review*, Vol. 19 No 1, January 1989):

“It is important to note that the nature of both the Internet and UNIX helped to defeat the worm as well as spread it. The immediacy of communication, the ability to copy source and binary files from machine to machine, and the widespread availability of both source and expertise allowed personnel throughout the country to work together to solve the infection even despite the widespread disconnection of parts of the network.

Although the immediate reaction of some people might be to restrict communication or promote a diversity of incompatible software options to prevent a recurrence of a worm, that would be entirely the wrong reaction. Increasing the obstacles to open communication or decreasing the number of people with access to in-depth information will not prevent a determined attacker—it will only decrease the pool of expertise and resources available to fight such an attack. Further, such an attitude would be contrary to the whole purpose of having an open, research-oriented network. The Worm was caused by a breakdown of ethics as well as lapses in security—a purely technological attempt at prevention will not address the full problem, and may just cause new difficulties.”

What hath DoD wrought: Net Results

The ARPANET is often nicknamed the “Grand-Daddy of Packet Networks,” and rightly so. (The more staid journals often say “progenitor.”)

Twenty years later, ARPA’s modest experiment has yielded a legacy almost too mind-boggling to grasp. For people in the computer, research, engineering and other technical communities, the ARPANET’s impact explicitly permeates the work we do and how we do it.

continued on next page

The ARPANET is Twenty (*continued*)

One way or another, the ARPANET, its direct and indirect children, and its by-products touch a startling number of the industrialized population.

Uses Packet networks carry our credit card authorizations. They support private and public electronic mail networks, PDNs, VANs, and bulletin boards, whose total populations run into the millions. They provide the communication backbone for universities, for government agencies, and multi-national corporations. (See "Children of the Net," p. 9).

The list of products, organizations, and concepts that have come out of the ARPANET is equally legion. As one example, TCP/IP, which became the foundation for UNIX-oriented LANs. Equally, consider the network protocols of IBM, DEC and most other vendors, whose installed networked computers serve uncountable millions, owe substantial debts to the ARPANET project.

Want more? How about gateways? Routing? Electronic mail? Network protocols, paving the way for OSI. Host protocols. Public data networks. Topological network design tools, including expert systems. Protocols for inter-program resource sharing. Remote computer management. On-line RFPs, and the concept of other on-line information servers and services. Multiprocessor architectures. Yes, the ARPANET had 'em first—the BBN *Pluribus*, a good decade ago, serving as IMPs and gateways in a few networks. Packet radio. Satellite Packet. LANs—all "demand access" oriented applications. Packetized voice. Multi-media collaborative mail systems. Geographically dispersed simulation and training networks. Networks to link supercomputers. That's it, for starters.

So it's not hard to make a case that the history of the ARPANET is, to a remarkable degree, the history of modern data communications. For many of us—without a doubt, nearly everyone who reads this article—it has changed the way we work, think and live.

The ARPANET project has been so successful, in fact, that the original network has become tasked with responsibilities far in excess of its capability or purview—and will be, at least physically, a victim of its own success.

This nearly happened half a decade ago, as operational traffic grew inappropriately to the responsibilities of a research testbed. The ARPANET was 'rescued' by the resulting ARPANET/MILNET Split. As part of the Defense Data Network program, the ARPANET and Internet continued to expand with new bandwidth, new components, and new sites. This time, however, it looks like our old friend will die the "true death."

The future As a result of a combination of organizational restructuring and newer technologies, that which we have known as the ARPANET will over the next several years be absorbed along with several other government networks into the new *National Education and Research Net*, or NERN.

Networks scheduled under the NERN program include the *Defense Research Internet* (DRI), *NSFNet*, the *NASA Science Internet*, and the DoE's *Energy Sciences Network*.

According to Craig Partridge, a scientist at BBN Systems and Technologies Corporation and active member of the Internet Engineering Task Force, "The DRI will be basically an experimental network, without the goal of major connectivity or operational traffic. This R&D/testbed function was the ARPANET's original purpose."

The DRI is already testing T1-speed backbones, and slated to explore a new generation of technologies and concepts, from Gigabit-speed traffic to we cannot begin to imagine what else.

Other networks, says Partridge, like NSFnet and the NASA Internet will handle production traffic. "This is where the IP traffic and connectivity will be provided."

The key influence of the ARPANET, he adds, "has been to provide the source of many important ideas in networking, and the testbed to try them. The ARPANET forced us to think about networking, and gave us a place to try them out—the majority of key ideas in networking, in my opinion. And we're taking all that accumulated wisdom to the DRI."

But meanwhile, the ARPANET is alive and well. So, Happy Birthday, ARPANET! As Gilbert & Sullivan say in *The Mikado*:

*This toast with three times three we'll give
Long life to you—till then!*

Written On The Net

This article is itself, it should be noted, a product of the ARPANET and its spin-offs. Much of the relevant correspondence, including confirmation text for quotes, went by e-mail over the ARPANET, Internet, USENET, or MCI Mail (including MCI Mail to MCI Fax). Some individuals were located via the ARPANET on-line directory service at SRI-NIC, using a *Telnet* connection. Portions of text came from documents initially *FTPed* from remote hosts. Text for the article itself was sent to *ConneXions* via the net. [Ed.: as are 97% of all submitted articles to this publication.] Some quotes and text may never have existed on paper, prior to this newsletter's final production and printing.

Children of the Net

"Are you on the net?" If that's the question, "the net" means the *Internet*, and "on" means "can we exchange e-mail?" The Internet is the network of interconnected TCP/IP-based networks that has exploded around the ARPANET, links what was at last count some 850+ individual networks and sites in a vast electronic community. Internet members include universities such as MIT, UC Berkeley, Columbia, Rutgers, and dozens more; major vendors, such as Digital Equipment Corporation, which alone has over 30,000 hosts and 100,000+ e-mail users on its mail network, Xerox, BBN, MITRE, and others; plus numerous other networks and organizations.

continued on next page

news briefs . . .

NATIONWIDE NETWORK OF
COMPUTER CENTERS FORMED

The long-envisioned nationwide network linking computer centers and different makes of computers will go into operation before the end of 1969 with an initial subnetwork consisting of UCLA, with a Scientific Data Systems Sigma 7; Stanford Research Institute, with an SDS 940; the Univ. of California at Santa Barbara, with an IBM 360/50; and the Univ. of Utah, with a DEC PDP-10.

The project, which is backed by the Department of Defense's Advanced Research Projects Agency, ultimately will number 14 computer centers in the network by the end of 1971. The effort was proposed and is headed by ARPA's Dr. Lawrence G. Roberts. The system is designed to make available to all members of the network the programs, computer power, and specialties of each center. In the beginning four-member network, UCLA will specialize in network analysis; SRI will be the information center; UC at Santa Barbara's specialty will be speech recognition; and the Univ. of Utah will handle graphics.

Each center will be equipped with an Interface Message Processor (IMP), now being developed by Bolt Beranek & Newman, Cambridge, Mass., under a \$1 million ARPA contract. The IMP will operate as a message switching device, and will translate the various machine languages during transmission and reception to make computers with different word lengths compatible with each other.

The second phase of the project will take place in 1970 after an evaluation of the four-member system and will incorporate six more centers into the network. These are the RAND Corp., Bolt Beranek & Newman, MIT's MAC project and the Lincoln Laboratory, the Univ. of Illinois, which will utilize its ILLIAC IV computer, and Systems Development Corp. The final phase tentatively will add Carnegie Mellon, Harvard, Dartmouth, and Stanford universities.

Professor Leonard Kleinrock, who heads the project at UCLA, stated that the university would serve as the network measurement center, providing analyses of the system and mathematical models of the network for evaluation and determination of future procedures.

Reprinted from
DATAMATION, April 1969.
© Cahners Publishing Co.

Children of the Net (*continued*)

For purposes of e-mail, "on the net" also encompasses users on BITNET, CSNET, JANET, and USENET. The total connected population is impossible to gauge exactly, but half a million is a good lowball estimate. (And that's not even counting USENET). The importance of the Internet as a resource and community is easier to state. Many consider it essential—so essential that working somewhere that is not on the net is close to unthinkable.

Recently, the Corporation for National Research Initiatives has begun experimental mail gateways between the Internet and some of the Public Mail Data Networks, like *MCI Mail*, which in turn is already gatewayed to other PMDNs like *CompuServe*.

Can "World-Net," a near-global interoperating e-mail environment, be far behind? Stay tuned for developments.

Packet Switching in 200 Words or Less

Packet-switching is a form of multiplexing data in a way that lets data segments share common lines, rather than be fit into dedicated time or frequency slots on point-to-point links.

Each stream of information is broken up into a sequence of shorter, individually addressed and numbered messages called *packets*.

Dedicated network components called *packet switching nodes* (a.k.a. PSNs or nodes, originally IMPs) forward packet streams from source to destination, using the address and sequencing information in each packet.

When each sequence of packets reaches its destination, the receiving computer—a *Packet Assembler/Disassembler* (PAD), or a host computer with PADding software—turns the stream back into its original unpacketized information stream.

Because each packet has its "identity" built in, the packets from any number of messages can share a common network of PSNs, PADs, and communications lines. The packet network can provide connections between any sets of permitted end-points, and the packets from hundreds of messages, file transfers, interactive connections and other application activity can all flow together, just as letters, magazines and packages all share a common postal system.

DANIEL P. DERN is a Massachusetts-based independent writer specializing in technology and business. He also writes computer humor, musical theater, science fiction, and how-to's on consulting and PR. During his six years at BBN as PR Manager, marketing writer, and technical writer, he wrote about ARPANET and UNIX technology, including the DDN's Non-Technical Network Overview, and numerous feature articles for the trade press.

The ARPANET is Twenty: *Interview with Vint Cerf*

by Daniel P. Dern

One of the three plenary speakers at the INTEROP™ '89 conference, Dr. Vinton G. Cerf's career spans the creation and growth of the ARPANET. He was at UCLA for the installation of the very first ("IMP 1") node, played a major role in the growth of the ARPANET and Internet, and was responsible for developing MCI Mail. He is currently Vice President of the Corporation for National Research Initiatives.

The following is drawn from a July, 1989 telephone interview with Dr. Cerf:

The first IMP

VC: I can remember when the first IMP came to UCLA. It was September 1, 1969. I hear they're bringing it to the INTEROP show floor—that's going to be a nostalgic encounter for a lot of us!

And now, twenty years later, we're looking at the dismantling of the ARPANET. But although the network itself will disappear, its legacy is firmly in place. What people do in data networking in the future cannot help but be strongly influenced by what we have learned, and by the technology spawned by the ARPANET.

DPD: What have we learned from the ARPANET experience?

VC: One, that the demand for useful technologies always grows faster than you can ever imagine.

We had thought, for example, in the earliest days, that six bits of address space would be enough to identify every node in the ARPANET. So the original network had a limit of 64 nodes.

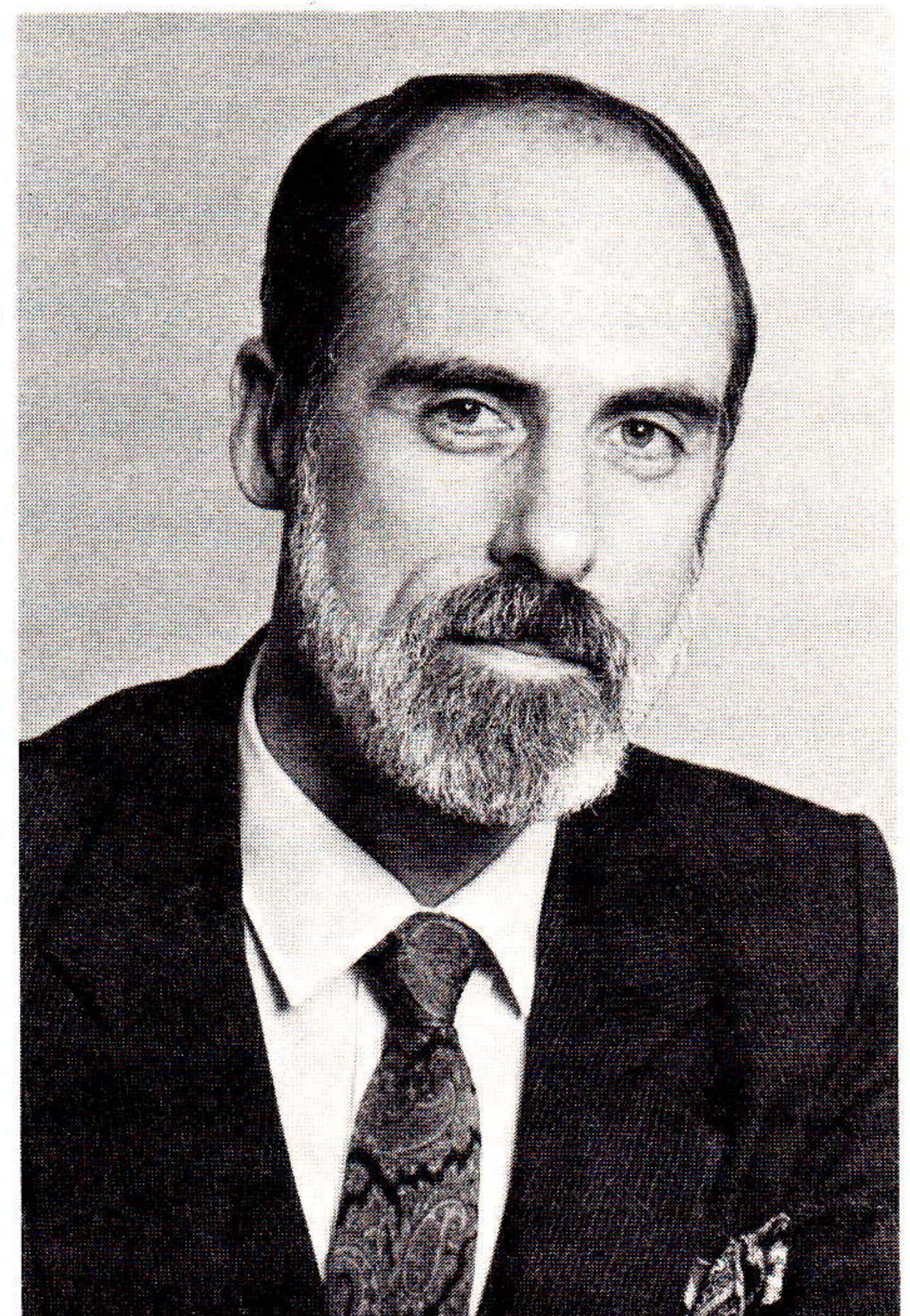
We soon discovered that that was not enough. We upped it by a factor of four. And we're now well past that number, in the MILNET.

Design for growth

So design your system with at least a growth factor of a hundred or so. In retrospect, even that wasn't enough for Internet. But in a period of twenty years, none of us would have guessed that what we now know as the Internet would be as big as it is today—and it is still rapidly growing.

Another interesting observation, looking back, are the things we thought we would do, with the ARPANET itself, that we never did.

The network began as what we thought was a project in *resource sharing*. We achieved some of that.



continued on next page

Interview with Vint Cerf (*continued*)

There was and still is a great deal of remote access to other people's timesharing systems. But there is only a modest amount of wide area networking program interaction, unless you include e-mail.

Electronic mail

This leads me to my third point: the incredible success of *electronic mail*. With the exception of Larry Roberts, none of us ever imagined that electronic mail was an application that would become so crucial to the community, once the network was in place.

Larry will claim, with good reason, that he could see that coming. In fact, as I recall, he was the first to write an e-mail application—using a TECO macro!

DPD: What are some of the technical lessons we've learned?

Routing is difficult

VC: ARPANET has taught us that *routing* is an incredibly hard problem. You can see that in the amount of routing research, and the incredibly dramatic changes in routing algorithms during their decade or so of existence. We should have had more evaluation and analysis done by people who have skills in control theory.

The first routing algorithms, based on hop count, were fundamentally unstable. IMPs often made calculations based on information of different "vintages," which could lead to routing loops.

These could be fairly severe. From 1977-1979, there were some severe routing "events" that caused a great deal of loss of connectivity. Things would get caught in these paroxysms, and roll around in the net, sometimes for hours, before they would stabilize.

It was only after a great deal of effort that the *Shortest-Path First* (SPF) algorithm currently in use was developed, which maintained a knowledge of the entire topology of the network.

We also discovered during this period the importance of line up/down decisions as perhaps the most fundamental decision in routing.

DPD: What were some of your roles in the ARPANET?

Performance tests

VC: In the earliest days, when the first four nodes were being installed, I was at UCLA. I ran something called the *Network Measurement Center* (NMC) for Len Kleinrock. The NMC was the computing facility used to load and measure performance of the ARPANET.

That was where I met Bob Kahn. He was the principle architect at BBN designing this network. He flew out to UCLA after the net was installed. He'd come with a new set of performance tests to do. I'd whomp up a program, to load the network, and we'd watch the network react and sometimes crash. Bob predicted two lockup phenomena: *store-and-forward* and *reassembly lockup*. No one believed him. We were able to induce this mode of failure by artificially loading the network. It was pretty exciting for a graduate student like me to have a hand in some of this.

While at UCLA I was also very much involved in the creation of the host level protocols. NCP, the predecessor to TCP was developed there under Steve Crocker's guidance.

We also worked on Telnet and FTP, on some work on RFC 733—one of the first standards for e-mail headers, replaced later by RFC 822. But my principal involvement at UCLA was in network measurement, which is down inside the net itself, and the host-host protocol.

Internetting

I left UCLA and went to Stanford, where I started a research program on internetting with the support of Steve Crocker and Bob Kahn, both of whom were at DARPA by then. That's where TCP was born. From 1973 to 1976, my research was focused almost exclusively on the design and construction of protocols that would let multiple packet nets link together.

Then I was asked to come work at DARPA, in 1976. I managed the internetting program and other packet technology programs, including the network security program, until 1982. During this time I was responsible for managing government involvement in internet technology, working with contractors from many agencies and organizations.

In 1982, I left DARPA for MCI. I was responsible for the design and construction of MCI Mail, which started in December 1982, and was up and running publicly on September 27, 1983.

Currently, I'm at NRI, which is a non-profit R&D activity funded by the government and by industry. Some of our principle projects involve the design of a digital library system, and design and development of Gigabit network technology, for the National Education and Research Network.

E-mail experiments

We are also researching inter-organizational electronic mail by experimenting with the linking of commercial public e-mail systems with the Internet. We have a link up between MCI Mail and the Internet, and we hope to add other commercial carriers to that over the next few months.

DPD: It sounds like you've been continuously involved in packet networks.

VC: Yes, although my focus of attention from about 1982 on has been on the applications of computer communications rather than in the guts of the networks.

DPD: Where do you see it going?

Several standards

VC: Several things are happening. We are clearly no longer able to say there will be one single multi-vendor standard for network communications. I wish it were true; I think OSI is aimed at that. But we're going to have to position the Internet to handle several protocol suites: TCP/IP, OSI and probably DECnet, switching these packets at the router level.

Faster networks

Second, we can already see the shape of the Gigabit network technology. It's very close to being available for experimental purposes. We have fast packet switching running at 1.5 Gb/s per channel. We will see some of that technology by early 1990.

So the Internet architecture will have to stretch, to accommodate operations some 30,000 times faster than when it started.

continued on next page

Interview with Vint Cerf (*continued*)

What we *don't* know is whether the current protocols available—OSI, TCP/IP, DECNET or SNA—are suited to running at Gigabit speeds. We have to find this out in the next year. Will we have to invent a new set of protocols?

Some people have already concluded we do, so you hear about the *eXpress Transport Protocol*, XTP, coming out of Protocol Engines Inc, specified by Greg Chesson from Silicon Graphics who has devised a chip set to run this XTP. Interestingly, it can also run other protocols, so you may see some existing protocols tested; perhaps some Gigabit TCP/IP experiments in the next 18 months.

DPD: In addition to the technology, what are some of the changes we've seen, sociologically?

VC: First and foremost, that e-mail has become an important part of the R&D community, in how we work. Facsimile is becoming more common, but it's limited—I'd hate to have to edit information in a fax file...

But the existence of digital forms of communications have dramatically changed the way we do business. It's like overnight mail; there wasn't much demand until it was created. And now we are seeing an increasing pace in the reduction of delay in exchange of information.

DPD: What are some of the directions network technology will take us next?

The future

VC: Collaboration technologies, once we understand how to do it. We're just on the edge of creating shared workspaces for geographically dispersed people to work in. I think we will see this become a fundamental part of our work environment, within several years.

Network management is another area still wide open. Within the ARPANET itself, we've done well. But it's only now getting attention in the Internet, and it is still a major subject for development.

Hear Vint Cerf talk about the Future of the Internet Protocol Suite at



The Plenary is at 9:00am on Friday, October 6, 1989.

The ARPANET is Twenty: *Quotes from some of the players*

In the following interviews and shorter quotes, representative founding and key members of the ARPANET and Internet communities (plus a representative user or two) respond to the questions:

- What was your role in the ARPANET?
- What has the ARPANET meant to you?
- What have we learned from the ARPANET experiment?
- Do any particular incidents or anecdotes come to mind?

—Daniel P. Dern



*Robert Kahn,
President
Corporation for National Research Initiatives
Reston, Virginia*

kahn@isi.edu

“I was one of the principals in the design and implementation of the communications sub-net—the packet-switching system—while I was at BBN. The overall project involved people from all over the country, and took place over a period of several years. The initial development of the subnet occurred from January 1969 through August 1969, when we delivered the first IMP to UCLA.

ICCC

I was also directly involved in planning and carrying out the first public demonstration of the ARPANET, in October 1972, at the first *International Conference on Computers and Communications* (ICCC), in the Washington Hilton Hotel.

This was the first public unveiling of the net—seen by more than 1,000 attendees over several days. It involved approximately 40 terminals accessing many large computers connected to the net.

In late 1972, after the network was up and running as a nation-wide entity, I joined DARPA, where I subsequently became Director of the Information Processing Techniques Office, which had funded the original work on the ARPANET. While at DARPA, I played a lead role in advanced packet communication technologies as they related to ground-based radio systems and other media, and was also involved in other network issues such as end-to-end security and packetized speech.”

Q: What have we learned?

Electronic mail

“I think the first thing we’ve learned is the importance of computer networks—certainly to the computer research community—for the connectivity and the interaction it provides. One of the unplanned but vitally important developments was *electronic mail*, which provided the ‘glue’ that allowed the the research community to interact over the network.

We’ve also proven beyond a doubt that packet switching is a viable technology. The next major challenge is making it work in the Gigabit rather than kilobit world.”

continued on next page

ARPANET is 20: Quotes from the players (*continued*)

Q: Do any particular anecdotes come to mind?

"Probably one of the most striking, in its way, was when we were in the first stages of building and debugging the network, which had about ten nodes at the time. We could tell—from our remote facilities—where problems were occurring in various parts of the net, such as errors in phone lines. This was before the phone companies had similar facilities in their own central offices.

Broken line

I remember calling from Cambridge to a telephone company central office in California to inquire about high error rates on a certain line. First of all, the telephone company didn't understand why someone from Boston was calling about a line in California. And when we explained why, they said 'How could you possibly know what is happening on lines across the country?' And three minutes after I called, the line actually 'broke.'"



Heidi Heiden,
Senior Vice President
The Wollongong Group
Falls Church, Virginia

heidi@twg.com

"Back in 1981, when I was working for the Department of Defense, I was told to develop a potential alternative to the then-current AUTODIN II plan. This project had been under development for five years or more, and was under increasing scrutiny as to whether it would be viable, how much it would cost, etc.

What we envisioned when designing the *Defense Data Network* (DDN) was something more than simply a collection of lines and switches. We approached it from the perspective of solving a user-to-user connectivity problem.

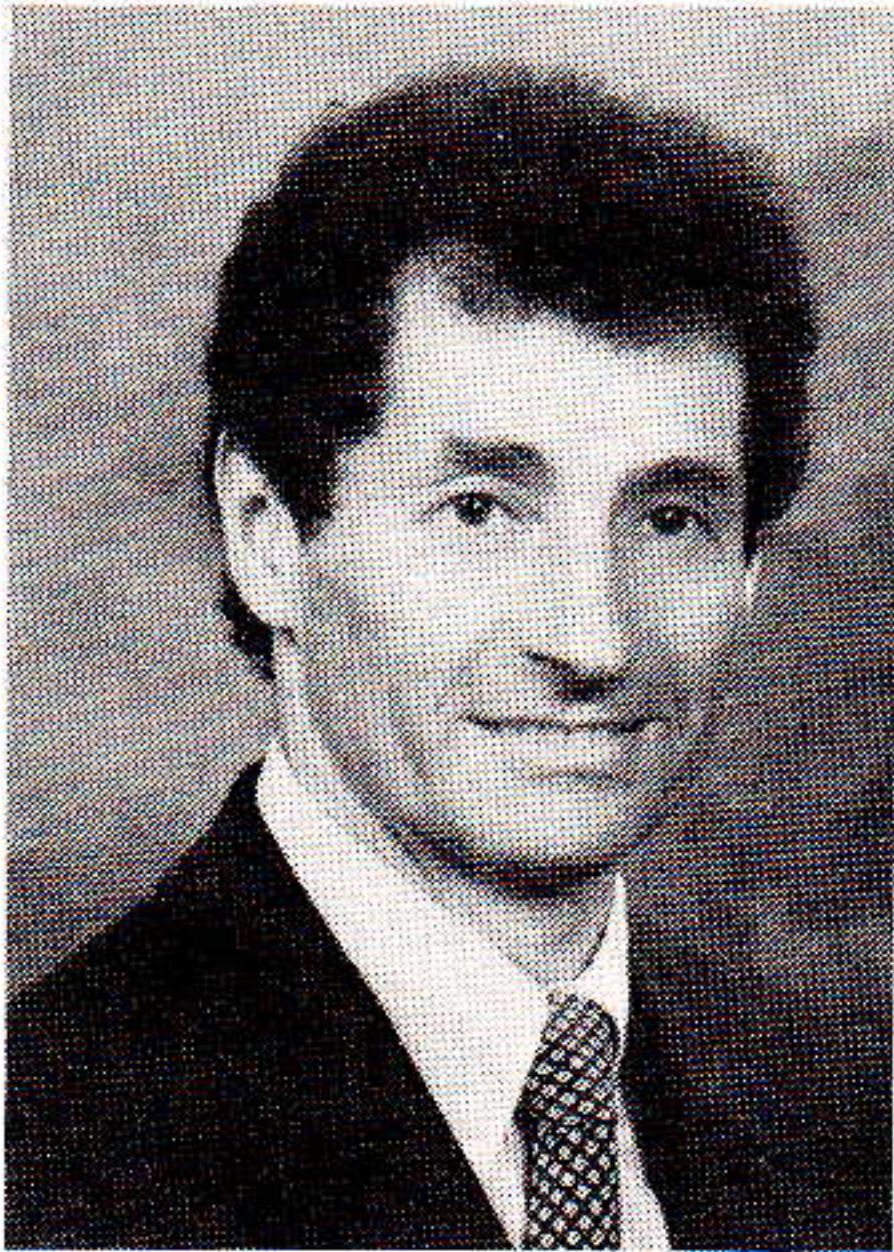
A protocol problem

Therefore, we had a *protocol* problem, *not* a network problem, because it's protocols that make everything happen.

I don't like reinventing wheels. We looked at the ARPANET, and saw a technology already in place and working, that did a lot of what I thought a Defense Data Network should do. That was the famous 'ARPANET-AUTODIN Shoot-Out.' ARPANET technology won on a competitive bid.

But we also looked at NCP (*Network Control Program*), and concluded that it was not the right protocol for our needs. That led us to TCP/IP—which was not in an operational network at the time, with the intent to migrate to OSI when it became available. We grew the ARPANET, and eventually split it into two separate networks—the ARPANET and the MILNET.

Our biggest success, I think, was considering it more as a protocol problem than a networking problem. Time has proven our approach was a correct one. The DDN, ARPANET and Internet all support *open* protocols, end-to-end connectivity, and TCP/IP. We were one of the first to stress open systems, and you can now see that architectural structure everywhere."



*Leonard Kleinrock,
Professor
Department of Computer Science
UCLA
Los Angeles, California*

lk@cs.ucla.edu

Measurements

"When Larry Roberts went to DARPA, he called on me to help create a network, particularly to define how to measure the performance. I had a lot to say about what 'hooks' should be put into the packet switching software so we could do measurements.

When the first IMP arrived, at UCLA everybody was ready to point fault-fingers at everybody else. Fortunately, it worked. We had bits moving back and forth the first day, and messages the next day. It was quite an event.

Our measurement studies proved to be very important. We found serious problems with flow control and routing procedures. We identified a number of things that needed improvement, like message reassembly techniques, and adding more buffer storage in the switches."

Q: What have we learned?

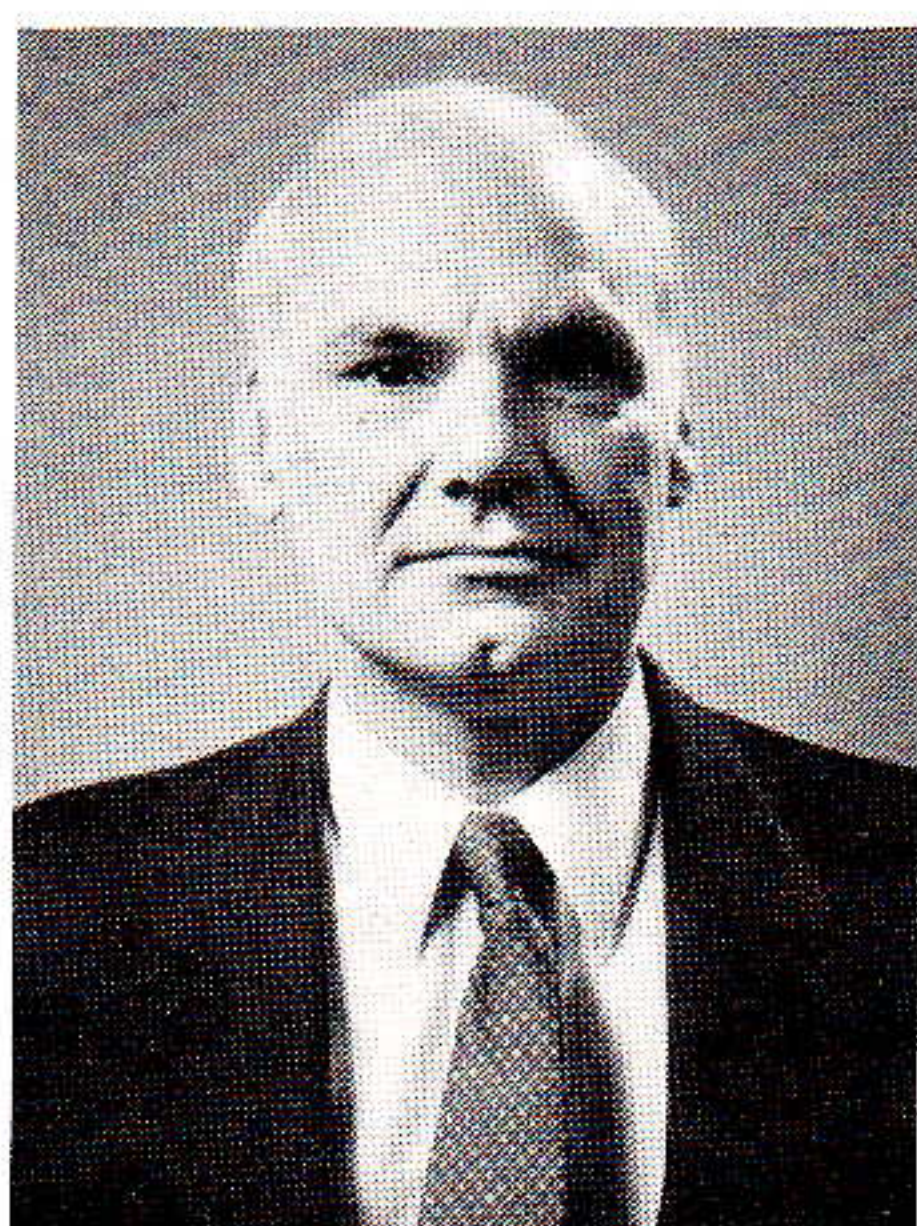
"Lots of things. First, the early experiment proved that packet switching *worked*. Secondly, we showed that a distributed routing control mechanism worked.

The efficiencies obtained by packet switching have proven to be what we expected, and continue to be. X.25 coming in was a phenomenally rapid development by the folks in the business. Larry Roberts was a major player in getting it through the CCITT.

Networking is *hard*

We discovered that networking is *not* easy. A lot of organizations have gone bust doing it. AT&T closed down *Net/1000* in 1986, after losing a billion dollars on it. Making money in this business is difficult. It took fifteen years after the ARPANET started before Telenet began making money. Networking is a service; it isn't useful until you have wide geographic coverage."

=====



*Larry Roberts,
President
Net/Express
San Mateo, California*

"I started working in 1962 on the concepts that proved fundamental to the ARPANET. I did experiments at Lincoln Labs in telephone interconnection, to see how we could make computer communications work. It worked very poorly, but we could do it. Already in 1963, an economic analysis showed that computing was getting cheaper much faster than communications, so we wanted some way to share computing resources. I decided we needed better communications to make this happen.

New concepts

And so I worked on a better concept. When I came to ARPA in 1967, part of why I was hired was to do that. What I wanted to do was to create the network technology.

continued on next page

ARPANET is 20: Quotes from the players (continued)

I wrote the first paper on this, in 1967, for the ACM Gatlenberg conference, outlining what the network was to be like, and defining that it should interconnect the major ARPA computer sites.

In addition to the packet switches, which BBN did the software for, we put in a lot of committee work on related aspects like design of host to host protocols, and getting the host computers involved.

The TIP

Soon after the network began operation, we realized we needed something that would let individual users access the net other than through their host computers, so we built the TIP—*Terminal Interface Processor*—a predecessor to those devices we now call PSNs.

Telenet

The network grew very quickly, to where in 1973 I was encouraging people to go into the data network business. The FCC was seeing this as an open possibility for new carriers. But AT&T didn't want to. So I helped form Telenet, and became president.

When we built the Telenet commercial network, we changed to a virtual circuit (X.25), which was more economical. And all commercial business has gone that way since.

The government and research networks have stayed with the older technology—*datagrams*. I believe that datagrams are far less economic than virtual circuits. We went through this huge transition to get people from circuit switching to packet switching. They did that, but couldn't make the next step to virtual circuits.

LANs

Simultaneously to the development of ARPANET, we were exploring how to handle packet in other environments and media. The whole set of theories for LANs built up in 1970-1972, when we were all excited about understanding packet technology.

Xerox/PARC was one of the groups working with us on the network. Bob Metcalfe, who was there, created Ethernet while he was there. He soon went on to form 3Com, and you can see the influence of the ARPANET's focus on interoperability and resource-sharing in what they do today.

By the time I left DARPA, the ARPANET was being turned over to DCA. The experiment itself had been done. But the network didn't stop.

FAX

I went to GTE when they bought Telenet. It was around this time I got interested in facsimile as the next emerging communications technology. I believed that the use of fax would change radically as quality improved.

Aside from image quality, the historic problem with fax was its efficiency and reliability. Packet switching was the obvious answer to the first, and private networks for the latter.

So I started a new company, *Net/Express*, as a subsidiary of DHL, to build up facsimile technology based on packet data networking.

We build facsimile interface equipment and network equipment to interface facsimile and packet switching equipment. As one of our first efforts, we built, and continue to operate, an international facsimile network upon which DHL is a customer.

We've developed equipment which allows you to convert Group 3 to Group 4 facsimile and back, since data networks really want to operate with Group 4 technology. So we provide interfaces, like a PAD, to a packet network, so we can carry traffic at about 30 times less expense than over a voice network.

And we do store and forward, multi-addressing, all the added value-added functions you need.

You can see how all this follows from the lessons of the ARPANET re packet switching and electronic mail, and my subsequent efforts in applying these technologies to business requirements."

Q: Vint Cerf says you were the first to forecast the tremendous use of electronic mail on the ARPANET.

E-mail "I just happened to get enamored with electronic mail, when I was starting the network, and realized how important it would be to people in their business.

I wrote the first program that would list on your screen all the messages you had received, so you could see what mail you had gotten without reading it all sequentially.

Later, I concluded that e-mail would not dominate business so much as fax would, because people needed a standard, and needed something they could use easily. We're integrating the two technologies to some degree now.

FAX versus E-mail But if you look at the transmission market today, as an indicator, you see that about three billion dollars a year is spent on facsimile service, versus about only two hundred million on electronic mail. That's over a factor of ten difference. Mind you, these figures are only for service, they don't reflect equipment sales.

But because of the tariffs and because pricing is partly a function of total volume, it is currently cheaper to send a page of fax on the voice network than it is to send an e-mail page on a PDN like Telenet. And it's cost per page that drives the market."

Q: What have we learned from the ARPANET?

"The first thing we learned from the ARPANET is that packet switching worked, and that it was a far more attractive means of communication, with more flexible bandwidth, than circuit switching, for data.

Subsequently, we've learned that packet switching is also a better method for voice, although we haven't applied this yet.

Resource Sharing Second, we also showed in the course of the first few years of the ARPANET experience that computer power could be shared, nationwide, and that it was considerably more economic than everyone having their own mainframe. That's obviously become less important, due to minicomputers and micros. But at the time, we saved a factor of three in our computing bill. And if you look at the networks being done to access and share the NSF supercomputer centers, you see it's still a very strong motivation. Plus, sharing resources—files, etc.—is still important.

continued on next page

ARPANET is 20: Quotes from the players (*continued*)

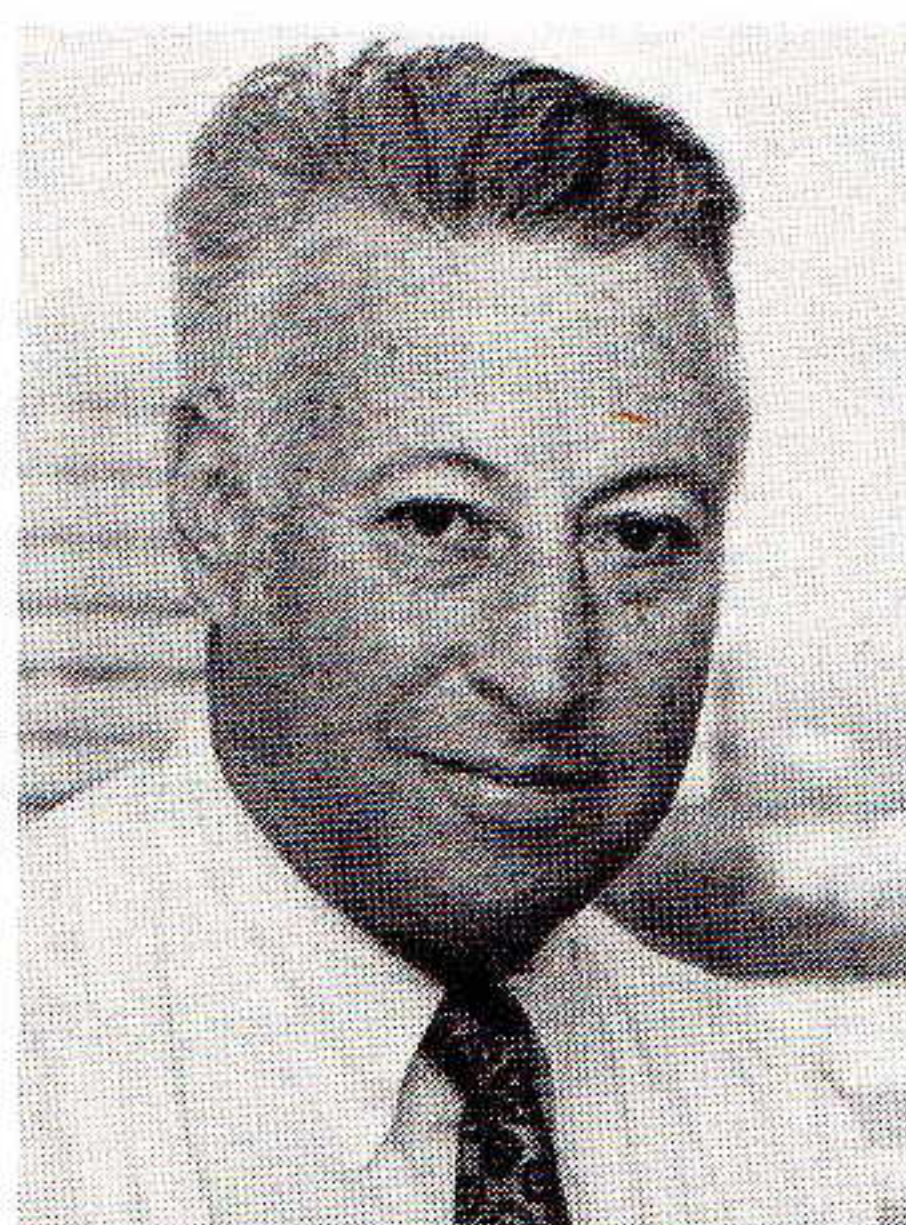
Thirdly, we learned how to do electronic mail, which is one of the biggest outcomes, and how much people need efficient forms of communications.

We had a very active period in the theoretical research. We used the net for electronic document distribution, on-line archives, and the like. People were publishing on-line through SRI—all the technology for satellite, radio, LAN and packet technology, all the ways to use packets were developed by the theorists like Len Kleinrock, myself, etc...

Yet when I started working on the concepts for the ARPANET in 1967 and 1968, people in communications in the government believed it was totally crazy, and would never work. It was only after it was working and had proven it would work that they finally changed their mind.

Reluctance to change

So one thing it's proven to me is that people find it very hard to change. There's a continual reluctance to change. We saw it in circuit versus packet, virtual circuit versus datagram. I expect to see it for packet voice and for video."



*Frank Heart,
Senior Vice President
BBN Systems & Technologies Corporation
Cambridge, Massachusetts*

heart@bbn.com

=====

Q: You were project leader for the original IMP software group...

"We had a team of people that was at that time probably one of the most knowledgeable groups in the world on how to connect computers via phone lines for real-time work.

We found a cooperative group at Honeywell to help with the hardware specials. We bid on the contract, and won.

That was a very very exciting time for those of us in it. It was like riding a technological rocket. It was a high point, for us at BBN and elsewhere.

The "IMP Gang"

The people who had a large role in writing the proposal and were responsible for the primary ideas, included Bob Kahn, who was at BBN at the time, plus Severo Ornstein, Will Crowther, Hawley Rising, and Dave Walden." [Now head of BBN Systems & Technologies Corporation].

Q: You've been quoted as saying that the ARPANET is one of the more important technological creations of the last few decades.

"We've created a whole industry. Its impact was surprising to those of us at ARPA, BBN and elsewhere who started the project. ARPA was viewing it as a demonstration project, and in remarkably little time, people were using, and depending on as a resource, what began as an experiment. It became part of our lives."

Social lessons

Q: What have we learned?

"There are social as well as technical lessons. One thing we've learned, is that if you keep the amount of government and corporate bureaucracy low—and get somebody who has strong control over the resources, like Larry Roberts did—you can accomplish a lot. We've learned you can have major successes with modest amounts of money, if you put the control in the hands of very bright people and let them work.

And we've learned that it was possible to build a packet switch. A lot of people didn't believe it would work, that you could do adaptive routing.

Reliability

Another important observation, which has been pointed out by people like Jay Forrester, that unless you put reliability as the first priority, beyond cost, beyond schedule, and beyond performance, things won't be reliable. In the IMP project, we put a tremendous emphasis on whatever we could do to make it reliable. We even ruggedized the first units. We worried about cross-patching, power recovery, remote debugging. We set a watchdog timer so that if something went wrong in the program it would reload itself. There were no switches or buttons used in normal operations. The remote control part in particular was very novel for the time.

One thing that stands out: the excellence of the groups that participated in the net. The network project attracted a coterie of very talented people, everywhere.

The project was also a birthplace for other critical technologies. BBN built the Pluribus, for example, a very early multiprocessor, and made many dozens of them for government networks. The ARPANET was a spawning ground for technologies—host protocols, ISO—and gave a major shot in the arm to computer R&D. And NSFnet's using networks to provide access to supercomputers is a clear outgrowth of ARPANET."



Einar Stefferud,
President
Network Management Associates, Inc.
Huntington Beach, California

stef@nrtc.northrop.com

"I was an interested bystander from 1968 till 1975 when I became an active user. Perhaps Hyperactive, as I was immediately struck by the potentials for the technology, especially netmail. My entry to the ARPANET was under a series of contracts with the US Army Development And Readiness COMmand (DARCOM). My first significant contributions were through establishment and operation of *MsgGroup*, one of the first of many ARPANET mailing lists."

Q: What have we learned/gotten out of ARPANET?

Social infrastructure

"The ARPANET has produced several monumental results. First, it provided the physical and electrical communications backbone for development of the latent social infrastructure we now call *The Internet Community*. The ARPANET provided the 'developer' and now, after 15-20 years, we have a truly magnificent social infrastructure.

continued on next page

ARPANET is 20: Quotes from the players (*continued*)

It of course has all the strangeness and foibles of any other social infrastructure, but it also has a variety of critically important properties. Among other things, it has taught a large number of people how to work together with fast moving information. I suspect that the basic metabolism rate for new idea formation and development is much higher in the Internet community than elsewhere, but I must admit that I have no data. Maybe someone can figure out how to do some research on this question.

OSI Another thing that flowed from the ARPANET was the basic technology for Open Interconnection of Systems, which is now called *OSI*. (Unfortunately, the emphasis of the acronym is on *Systems* rather than *Interconnection*—Just one of the problems with the use of acronyms!)

The contributions to OSI are not limited to proving that IP works. If you dig around, you will find that much of the seminal work that led to adoption of the upper-layer architecture of the ISO/CCITT standards and recommendations for OSI flowed from research done in the mid 1970's in the ARPANET community."



*Eugene H. Spafford,
Assistant Professor
Department of Computer Sciences
Purdue University
West Lafayette, Indiana*

spaf@cs.purdue.edu

"We're just beginning to realize the potential for communications and dissemination of information that technology like the Internet brings to us.

William Wulf, head of computer and information science and engineering at the NSF, along with other folks there, are working on something they call the *Colaboratory* that will use network technology to let people in a shared discipline access the same data and work on the same projects as if they were in the same location. It's an exciting idea. It requires more bandwidth than the Internet currently has. But they're working toward that goal.

From my experience in the USENET, which is loosely related to the Internet, we've seen some very interesting things occur.

A unique forum for communication

When you're relating to other people through the medium of electronic mail—the discussion lists, the forums, and direct exchanges—all you know about them is what they say, in a very positive sense. You have no inherent way of knowing if they're young, old, male, female, short, tall, Jewish, Gentile, Muslim, physically or sensory handicapped, etc. And that can be an intriguing, enabling concept. It provides a unique forum for shared discussion and development. At the same time, we're continuing to see problems with this evolution and expansion of electronic culture. We've reached the stage where we need new forms of electronic social etiquette—*netiquette*, as some call it.

This isn't so much a hardware or software challenge, although programs and protocols may help us out. It's a *social development*, and it's something we'll need answers to, if our networks and our uses for them continue to grow."



Geoff Goodfellow,
President and Founder
Anterior Technology
Menlo Park, California

geoff@fernwood.mpk.ca.us

"I was previously at SRI International for 12 years and have been an ARPANET resident since 1973. What the ARPANET has meant to me is being on the cutting edge of networking and communications technology.

Failures

From my perspective, I'm interested in the *failures* and shortcomings of the Internet as it has evolved.

What began as a single network developing packet-switching technology, has turned into an operational amalgamation of well over 800 individual networks, sharing a T1 backbone, and already looking at T3 speeds (45 Mbps).

There has been no management of the Internet's growth. The Internet is lacking in 'hands-on' guidance and direction. There has been a paucity of software tools, or mechanisms, to manage the explosive growth of the Internet over the past few years. We love to throw faster processors and more bandwidth at performance problems, instead of looking at how we should be growing the network and using extant resources efficiently."



Jack Haverty,
Chief Architect
BBN Communications Corp.
Cambridge, Massachusetts

haverty@bbn.com

"At BBN, I've been involved in how the networks worked, how to build them, particularly in the Internet—projects like making the first gateways work in an operational, field environment.

What have we learned? As an analogy, think of electricity. Not much was done with it until somebody invented light bulbs. The ARPANET is like that: it didn't take off until people started figuring out how you could use it to do real end-user *applications*, like remote login and moving files around, and new applications like e-mail. E-mail was the first fundamental application that was enabled by the network—something you couldn't do just with wires.

Another important lesson we've learned, particularly in the past ten years, is that taking a technology from an R&D environment into an operational environment is a lot tougher than it sounds.

Going from a small scale network operated by its developers into a technology which can be deployed and supported in a large scale was and is a significant technical challenge.

continued on next page

More development needed

ARPANET is 20: Quotes from the players (*continued*)

It's still more an art than a science. There's no handbook of network engineering you can go to for a formula. We're still standardizing things like network management and protocols for the Internet. Entire concepts, such as accounting mechanisms for producing 'bills,' need to be invented post-hoc.

And to continue the electricity analogy, we don't yet have much in the way of inspectors, UL codes, circuit breakers and all the ancillary technologies and procedures we need.

But the good news is that reliable network systems can be fielded today. There is still a lot of work to be done to turn the art of inter-networking into an engineering discipline, but that's why this whole area remains interesting and fun."

=====
Ellen Eades,
Technical Writer
Microsoft
Redmond, Washington

microsoft!ellene@beaver.cs.washington.edu

"As a technical writer at Microsoft, I see electronic mail used very heavily, to keep people apprised of software development activities, and how documentation must be changed. E-mail is also used simply to facilitate employee communication. I use the USENET for communication with other people whom I will probably never meet.

Contacts

The net has given me a circle of contacts from all over the United States and Europe. I participate in a number of non-technical newsgroups on USENET, and correspond regularly with people whom I have never met in person on subjects ranging from folk music festivals to feminism."



=====
Chuq Von Rospach,
Senior Technical Support Engineer
Apple Computer
Cupertino, California

chuq@apple.com

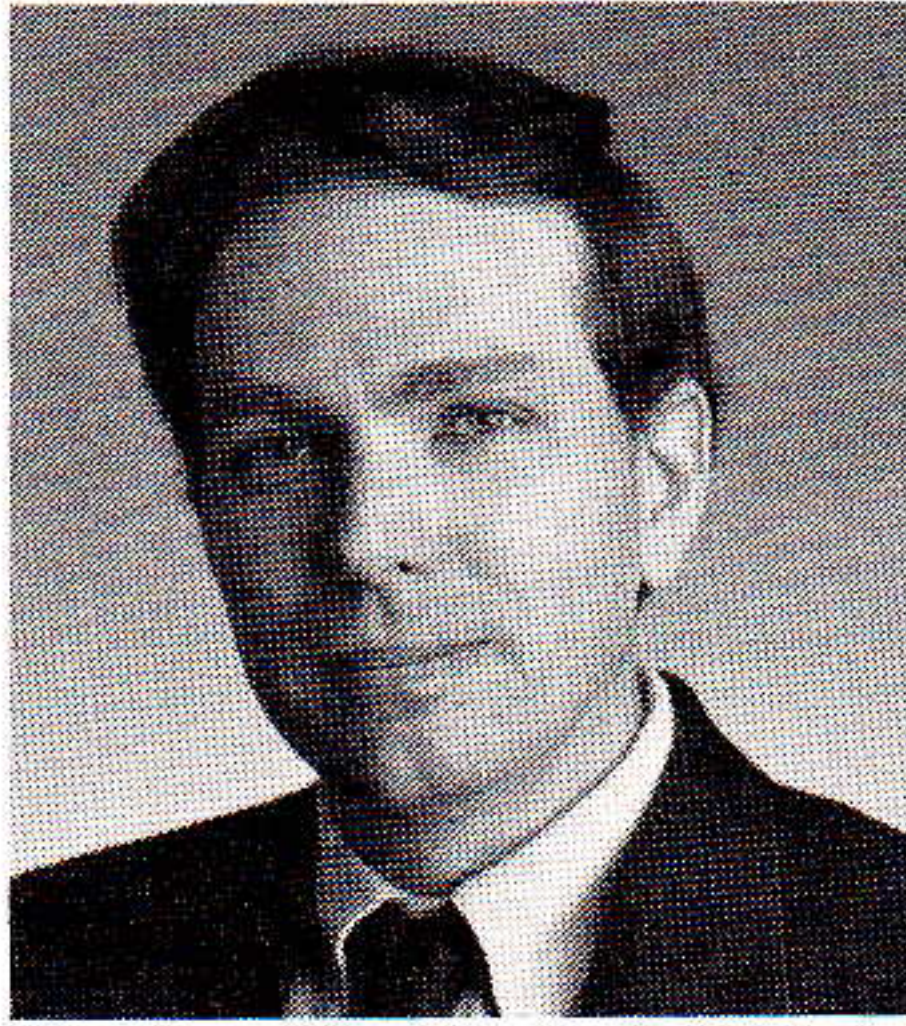
"I've always been primarily a user. I started out as CHUQUI@MIT-AI.ARPA back when MIT still accepted open accounts. From my point of view the ARPANET has primarily been a medium for speeding up what was already going on and making it happen better. It's also allowed people in different areas of the world to collaborate on a project effectively without the lags and delays and information loss that this would normally imply trying to get data from one area to another using something other than computers."

Q: Do any particular incidents or anecdotes come to mind?

The Worm

"The *Internet Worm* was a major one. I was in charge of getting the worm under control for Sun—not isolating or fixing it, but making sure the customers knew about it and got it fixed. By far the worst week of my life. It brought into reality a point anyone involved with computers knows is a possibility—exactly how vulnerable anything that relies on computer technology is to outside forces. That's *scary*."

=====



Steve Kent,
Chief Scientist
BBN Communications Corp.
Cambridge, Massachusetts

kent@bbn.com

Network Security

"I started as an ARPANET user, back in the early-to-mid 1970s. By the late 1970s, I was involved in security projects for the ARPANET and Internet.

There were two major motivations for network security. One, some of the perspective users of this networking technology had sensitive information, and they couldn't use this technology unless there was a way to protect their information.

In the larger context, there was the feeling that we ought to understand how to provide various types of protection for subscribers, who might not have classified information, but wanted some level of protection.

I participated to a certain extent in the old TCP/IP working groups, which had 40-50 major contributors. I eventually became a member of the *Internet Configuration Control Board*, which was predecessor to today's Internet Activities Board. One of my major roles in that context was to monitor developments in protocols from the security perspective, to understand that a particular functionality being proposed might pose security problems, make suggestions for more secure protocols. That evolved over time into my role as the chair of the *Privacy and Security Task Force*.

While I've been involved different parts of the Internet technology, from transport protocols to voice to gateway protocols, the common theme has been the security aspect."

Q: What have we learned?

"I think that the actual benefits that have come from this experiment were greater than originally envisioned, but different.

In the early days, there was a lot of hope that people would implement loosely coupled distributed systems employing the ARPANET. There were early efforts along that line, like the *National Software Works*, in which Bob Thomas was a major player, for DARPA, where we envisioned making use of different computers for compiling, databases, etc., gathered around the network.

For the most part that didn't happen. Partly because communications technology in the wide area environment wasn't fast enough, and because the state of software engineering wasn't advanced enough to make it practical. That's changed more recently, with things like high-speed LANs.

Electronic mail

What the ARPANET did was provide this ubiquitous *electronic mail* facility, which may have been its single greatest qualitative impact. The Telnet remote login facility was also a boon to many folks, but that was not a qualitatively different facility, just more economically attractive than dialup access through the phone network. But within the research community, what we really gained was electronic mail, as a means of communication among the participants in the experiment. And that is something that really caught hold. It significantly enhanced productivity.

continued on next page

ARPANET is 20: Quotes from the players *(continued)*

Mailing lists

It gave rise to new cultural phenomena—the idea of affinity mailing lists that cropped up, from TCP-IP to computers and society. That was important. It facilitated productivity, it is the one application most widely used across the Internet, and it touches the lives of Internet users most frequently. E-mail is very visible to the users.”

Q: Do any anecdotes come to mind?

Black Tuesday

“Global routing failures stand out in my mind, in particular, the first global routing failure in the ARPANET, in the early 1980s, often called ‘Black Tuesday.’

This event gave a real demonstration of the ingenuity of the BBN network operations staff to diagnose and deal with a very difficult problem.

It also gave an appreciation of the complexity of a distributed real time system of this magnitude. All those packet switches interacting with one another, running a fairly complicated routing algorithm made for a very difficult fault isolation and resolution procedure.

Eric Rosen wrote a good report on what happened, and why, how we recovered and what we did to prevent a recurrence.

It takes effort to figure out what has gone wrong, and then recover and bring back a network of such geographic extent—without sending somebody out to every site to fix it. It was pretty impressive.”

=====



*Marshall Rose,
Senior Scientist
NYSErNet
Western Development Office
Mountain View, California*

mrose@nisc.nyser.net

Technology frozen in time

“The Internet has been an wonderful environment for learning about internetworking. It has proven to be not only an excellent vehicle for doing research, but also for sharing research results. These results have been transferred to the users in a timely fashion—the Internet has served well both for research and service.

Even with these advances, the technology is much more static than most people appreciate: a lot of the technology seems frozen in time, and there is a great tendency to avoid incorporating or learning from alternate technologies, such as OSI.

Other than a student, my role has been to incorporate these different technologies into the Internet.”

Requiem for the ARPANET — by Vint Cerf

*Like distant islands sundered by the sea,
We had no sense of one community.
We lived and worked apart and rarely knew
that others searched with us for knowledge, too.*

*Distant ARPA spurred us in our quest
and for our part we worked and put to test
new thoughts and theories of computing art;
we deemed it science not, but made a start.*

*Each time a new machine was built and sold,
we'd add it to our list of needs and told
our source of funds "Alas! Our knowledge loom
will halt 'til it's in our computer room."*

*Even ARPA with its vast resources
could not buy us all new teams of horses
every year with which to run the race
Not even ARPA could keep up that pace!*

*But, could these new resources not be shared?
Let links be built; machines and men be paired!
Let distance be no barrier! They set
that goal: design and build the ARPANET!*

*As so it was in nineteen sixty-nine,
a net arose of BBN design.
No circuit switches these, nor net complete
but something new: a packet switching fleet.*

*The first node occupied UCLA
where protocols and measurement would play
a major role in shaping how the net
would rise to meet the challenges unmet.*

*The second node, the NIC, was soon installed.
The Network Info Center, it was called.
Hosts and users, services were touted:
to the NIC was network knowledge routed.*

*Nodes three and four soon joined the other two:
UCSB and UTAH came on cue.
To monitor it all around the clock
at BBN, they built and ran the NOC.*

*A protocol was built for host-to-host
communication. Running coast-to-coast,
below the TELNET and the FTP,
we called this protocol the NCP.*

*The big surprise for most of us, although
some said they guessed, was another proto-
col used more than all the rest to shuttle
mail in content flaming or most subtle.*

*When we convened the first I Triple C,
the ARPANET was shown for all to see.
A watershed in packet switching art,
this demo played an overwhelming part.*

*Within three years the net had grown so large
we had to ask that DCA take charge
to operate a system guaranteed
for R&D and military need.*

*Exploring other packet switching modes,
we built the first spread spectrum mobile nodes.
The Packet Radio, the mobile net,
worked on the ground and even in a jet.*

*Deployed at SAC and Eighteenth Airborne Corps,
the Packet Radio unlocked the door
to what we now know as the Internet.
The driver for it all was PRNET.*

*The Packet Satellite, another new
technique, was added to the net milieu.
And then to shed more light upon the dark,
there came the Ethernet from Xerox PARC.*

*To these we added yet another thing
from MIT: a local token ring.
We saw the local net techniques compound
until the list could easily confound.*

*The Internet foundation thus was laid.
Its protocols from many sources made.
And through it all the ARPANET grew more;
It was, for Internet, the central core.*

*The hardware of the net was changing, too.
The Honeywell was first, and then the SUE,
which forms the heart of Pluribus today
though where this platform sits one cannot say.*

*The next big change was called the MBB.
It emulated Honeywell, you see,
so one by one they modified each node,
by means of closely written microcode.*

*Now known as 30 prefixed with a C,
these nodes are everywhere from A to Z.
The European MINET too was full
of nodes like these from Mons to Istanbul.*

*The second Autodin was long desired
but once accepted instantly expired.
Then to the rescue rode the ARPANET!
And soon the MILNET by its side was set.*

*By Nineteen-Eighty DoD opined
its data networks soon must be aligned
with Internetwork protocols, to wit:
by Eighty-Three the TCP was IT!*

*Soon every host that sat on ARPANET
became a gateway to a local net.
By Eighty-Six new long haul nets appeared
as ARPANET its second decade neared.*

*The NSFNET and its entourage
began a stately national dressage
and soon was galloping at T1 speed
outdistancing its aging peer indeed.*

*And so, at last, we knew its course had run,
our faithful servant, ARPANET, was done.
It was the first, and being first, was best,
but now we lay it down to ever rest.*

*Now pause with me a moment, shed some tears.
For auld lang syne, for love, for years and years
of faithful service, duty done, I weep.
Lay down thy packet, now, O friend, and sleep.*

Book Reviews

Internetworking Computer Systems: Interconnecting Networks and Systems, by John McConnell, Prentice-Hall Advanced Reference Series, ISBN 0-13-473976-0, 1988, 318 pp.

Layered approach

"Internetworking Computer Systems" by John McConnell is an ambitious little book. In slightly more than three hundred compact pages of text he covers just about every major aspect of the field which the title claims, surely this is truth in advertising. His approach is, appropriately, a layered one which takes the reader (after some history and definitions) from low-level data-link and media access control up through application protocols and the big issues of internetwork design.

Networking by example

The theme of the book is definitely OSI and he provides good overviews of how the various layers are given form by the standards. ARPA protocols are often used to compare and contrast with the OSI suite. Elements of several other standards (e.g., XNS, SNA, CCITT) appear in examples throughout the text, some extensive. A good summary of the book's style is "networking by example," the text is rich with brief illustrations of how real network protocols implement the concepts described.

Status of OSI

The text is peppered with the author's occasional interjection of his wisdom concerning the material being discussed. In a small section titled "Caveat Emptor" he writes about the current status of the OSI protocols (page 30):

"Efficient implementation has not been explicitly addressed at all. The higher layers for interworking have to interact extensively with the local operating system as they carry out OSI functions. Again, the choice of implementation strategies may result in excessive costs and performance degradation. Moreover, some aspects of the specifications will cause an excessive use of resources. Vendors, of course, will be willing to sell additional hardware or new, improved versions of their software.

The OSI model has ignored two basic areas: management and security. These areas are beginning to be addressed, but nothing substantive has been developed to date. These major omissions are sure to cause future trouble. Users may have to create their own services and then retrofit to a later standard.

The cavalier treatment of performance and cost issues may impel users to 'customize' their OSI products or to violate the standards in order to achieve the level of operation they need. If this practice becomes widespread, the OSI effort will have failed."

Good introduction

The material is technical, but far from technically dense. A person with moderate technical background desiring a good overview of internetworking could easily absorb this book in a few hours. I would describe the ideal audience as people who might read *ConneXions* but feel they could use a little more background to better understand the material presented here. This book helps provide the organizational gestalt of internetworking which is so important to grasp before proceeding to more detail. It also serves as an easy introduction to the vast terminology of this field, and would be very useful to executives and managers who need to become network literate.

—Barry Shein

UNIX System Administration Handbook, by Evi Nemeth, Garth Snyder, and Scott Seebass, Prentice-Hall Software Series, 1989, ISBN 0-13-933441-6, 593 pp.

The System Administrator's dilemma

It is quite common for new administrators to become quickly overburdened with the numerous and often tedious chores required to effectively manage a UNIX system. Documentation is normally at first cryptic, and it's no secret that it continues that way, and there is usually no road map to help them along their path of discovery other than grunts of the local usually-too-busy gurus. Of course, this doesn't sound like your site, right? I bet your systems take care of themselves. For those whose systems didn't walk out of the box straight away, plug itself in, and cheerfully spout "login:," please read on.

As a previous manager of UNIX system administrators, I was often looking for training material that would help increase the effectiveness of my people. I was searching for the unrealizable ideal document that got beyond the "here's just what you've got to do" and stopped there, to one that offered more knowledge as to "why" hopefully colored by folklore and current common practice. Such a book would get many brownie points also, if it took a step back from idiosyncratic site-specific methods and told about ones that were more objective and broad reaching. And of course, numerous examples are still always a must.

Organization

I'm happy to say that this new book is the closest I've seen yet to satisfying my ideals for a UNIX system administration guide. It covers a rich offering of system management activities to a depth that, when coupled with the system documentation, provides the SA with most of the background they need to get a good handle on the task at hand. This book covers numerous topics organized in a chronological order that suits the installation of a new system. The authors say that "We like this organization because it tells you what you need to know *when you need to know it* and because we feel that it increases the value of each chapter as a reference material." The book is consistently organized in a thoughtful and uncluttered manner. Both BSD and AT&T derived systems are covered equally. Humor is pleasantly spaced throughout the well written prose.

Networking

I was afraid when first approaching the book, that I would be overwhelmed with the SA idiosyncrasies at the University of Colorado, Boulder campus. I'm pleased that they have presented their methods in more of an illustrative than definitive manner. They go well beyond expectations in their coverage of networking (configuration, programs, and daemons) than any other single document I have read. They even describe the basics of *sendmail* configuration file taming—an experience previously reserved only for graduates of computer science with a minor in address weaving.

Recommended

The book is well worth the price and I recommend it for any UNIX system administrator who is still anywhere on the learning curve. This book, the system documentation, and a book or course in shell programming should significantly help any system administrator along in their path to discovering how wonderful and challenging it is to take the reins of a UNIX system with confidence. Bravo Boulder.

—Mark Laubach

continued on next page

Book Reviews (continued)

X Window Systems: Programming and Applications with Xt, by Douglas Young, Prentice-Hall, 1989, ISBN 0-13-972167-3, 468 pp.

Structured as a progressive series of examples of X concepts and Xt features, Mr. Young's book presents several programming applications of Xt Intrinsics and several widget libraries with the goal of "(enabling) the applications programmer to create *easily* and *more efficiently* mouse-driven graphical user interfaces."

Approach

In the preface the author contrasts the (traditional) approach to learning to program in X with the approach undertaken in his own text. He remarks that the process of understanding the basic X model, programming with Xlib, the C language interface to X, and finally the use of various high-level toolkits (e.g., Xt), has the advantage of forming a thorough understanding of and facility to create X application code, with and without the facilities (and limits) of higher-level toolkits. However, the time required and the bulk of material to be learned is substantial.

The second approach, the book's, undertakes to build applications using the Xt functions at once and tie together the various threads of narrative that are sandwiched between code fragments, to produce a working grasp of X, the Xt layer and widget libraries. The author allows that this method makes it unlikely that the reader will fully understand what is going on and will have to take things on faith, however, less time is required before the reader is working on real applications.

The introduction to the X architecture and motivation for the particular programming model the X Toolkit offered in Chapter One is brief and unsatisfying. Passing reference is made to the X philosophy of mechanism over policy, networks and the client-server model.

Organization

Chapters Two and Three provide an introduction to Xt Intrinsics. There is a few pages of discussion of some Xt fundamentals: initialization of resources, invocation of the shell widget—the root of the widget tree hierarchy, creation of child widgets of various classes (e.g., scroll bars), management of descendent widgets, event handling and widget options (e.g., widget size in pixels). The treatment of such a diversity of process management topics in just over seven pages is not detailed.

Widget programming

An example, a variant on command-line echoing follows. Functional complexity in the form of example uses of event handlers, callback and actions are added progressively. There are example uses of widgets, scroll bars, title bars, menus and dialogue boxes. In route, the reader is shown how to compile the code with the necessary libraries, unfortunately these are treated as black boxes and a loader directive is as close to Xlib as we get to the underpinnings. The example *Makefile* is flawed—the "success" message is not passed as an argument to *echo*. This does not inspire confidence in the accuracy of the code that follows. Mapping user actions to functions via the translation manager and multi-threaded support of multiple application contexts (logical applications) in the same address space complete the discussion and example code.

In X, resources are windows, graphics contexts, fonts, etc. Resources are almost everything and Chapter Three devotes 20 pages to resource management. While this is substantially more space on a single (fundamental) topic than any other topic introduced thus far, it is not presented as a coherent extension of the earlier explanation of the X programming model, nor is it connected in a deep way in the material that follows.

The remaining 200 pages are organized into 14 Chapters, almost entirely devoted to programming with widgets. Event handling is revisited, and a chapter is devoted to raster images, graphics contexts and primitive operations, fonts, inter-client communications and composite widgets. Some of the earlier topics are revisited in greater detail.

Summary

In summary, the book presumes the reader's knowledge of a substantial portion of X's milieu—so much is left to the reader's imagination. For the reader with no present involvement in an X-literate programming culture, the unreferenced remains unknown, even unguessed, aided very little by the modest bibliography in the appendices. I found the book's piece-wise approach limiting—assuming I did understand each piece and could proceed to the following material, at the end I could only make minor variations on the set of examples learned. The work is not a reference and is not recommended. Until Adrian Nye's work-in-progress becomes available, the following 3 papers are recommended:

A Simple X.11 Client Program, or, How hard can it really be to write "Hello World"?, by David S. H. Rosenthal, USENIX Conference Proceedings, Winter 1987.

The X Toolkit, more Bricks for Building User Interfaces, or, Widgets for Hire, by Ralph R. Swick and Mark S. Ackerman, USENIX Conference Proceedings, Winter 1987.

Using the X Toolkit, or, How to Write a Widget, by Joel McCormack and Paul Asente, USENIX Conference Proceedings, Summer 1988.

Introduction to the X Window System, by Oliver Jones, Prentice-Hall, 1989, ISBN 0-13-499997-5, 511 pp.

Great introduction

A pleasant, informed and well-structured introduction to the X-Window System. Not a reference, and highly recommended! Oliver Jones' book is a high point in introductory technical writing. His Xlib tutorial presentation (MIT X Conference, January 1988) has been expanded in scope and depth. It provides the reader with an understanding of X as a programmer's vehicle for writing applications and (!) as a network application whose behavior is affected by the underlying network.

The book is highly recommended to the X novice. For programmers already X-literate, the author's perspective and in-passing remarks on minutiae may be illuminating. It is certain that reading his presentation of "known" concepts and practical details will improve the reader's ability to make the same points clearly. The single most common "learner's problem" that I've observed, setting the display environment variable, is not overlooked.

—Eric Brunner

The Telnet Protocol

by Barry Shein, Software Tool & Die

Introduction

Telnet [RFC 854] is a network application program which uses the Internet Transmission Control Protocol (TCP, [RFC 793]) to provide a remote terminal service. Telnet relies upon the underlying reliability of TCP, no additional data validation is provided. All communication between the local and remote Telnet process occurs on a single, two-way data stream with encoded commands interspersed with the data. There is support for both seven bit (ASCII) data encoded in eight bit bytes (octets) and full eight-bit binary encodings. The goal was to provide both a user to host and host to host communications protocol.

Basic Model

The model Telnet uses consists of three parts: First, a *Network Virtual Terminal* (NVT) which is an imaginary device providing a canonical, generic terminal interface consisting of a keyboard and printer (or screen) interface. Second, a method for negotiating *options* to change the behavior of either or both ends of the connection. Finally, a *symmetric view* of the terminal and host end.

The last component, a symmetric view of both ends of the connection, has the potential to cause non-terminating behavior since commands and acknowledgements may not be distinguished. To prevent this from happening several rules are specified such as forbidding one end of the connection from acknowledging a request to enter a mode it is already in.

The Network Virtual Terminal

Telnet defines a *Network Virtual Terminal* (NVT) which is a bi-directional device consisting of a keyboard (for input) and printer (for output.) The default NVT (before options are negotiated) appears as a very simple terminal with the following characteristics:

Data is sent as seven-bit USASCII within eight bit fields. Commands are encoded using byte values outside of this range. When scanning the data stream a Telnet program is only required to check for "Interpret As Command" *IAC* (255) which must precede any other command bytes.

Input is buffered at the local host until some locally-defined signal occurs (e.g., hitting the <RETURN> key.) When all data for output and all data from input has been processed the Telnet process must send a special "Go Ahead" (GA) command to allow further data to be sent. This last feature implies a half-duplex connection discipline.

The *Control Functions* are defined as:

Interrupt Process (IP): A signal to the remote Telnet that the other end wishes whatever program is running be suspended, interrupted, aborted or whatever equivalent is available at that end.

Abort Output (AO): A request to abort any output from the remote process until some logical point in the future, such as a prompt for more input.

Are You There (AYT): A way for the user to get some acknowledgement immediately from the remote Telnet that it is still listening.

Erase Character (EC), *Erase Line* (EL): A request to erase the last character or line of input typed, respectively.

Telnet synch

This is sent via TCP's urgent out of band processing to indicate to the remote Telnet that it needs to immediately scan for *interesting* Telnet commands up to the next *DATA MARK* (DM.) Interesting Telnet commands in this context are defined to be either IP, AO or AYT.

For example, the remote server may be too busy processing to scan new input. Because of this backlog of unprocessed data on the remote host, new input may be sitting in buffers locally waiting for TCP's flow control to allow it to be sent. In this case a command such as IP would not get through until the remote host finally is able to resume reading the data and interpreting the input stream. This may be too late from the user's point of view.

In this case the user can send (or cause to be sent, typically by typing some locally understood character such as ^C) the following sequence:

First, an IP (or other interesting) command is put into the normal data stream. Next, a DATA MARK is inserted via the TCP urgent mode send operation. Optionally these can be followed by some string which is meaningful to the remote application possibly terminated by its own DATA MARK equivalent.

The urgent mode will bypass any flow control condition and be sent as soon as possible thus waking the remote Telnet. The remote Telnet will then begin reading its input looking for the (now expected) DM and processing any interesting Telnet commands it sees as it scans. Until the DM is found all intervening data is discarded. After the DM is found the remote Telnet resumes normal processing of further input. A DM seen in any other context will be ignored.

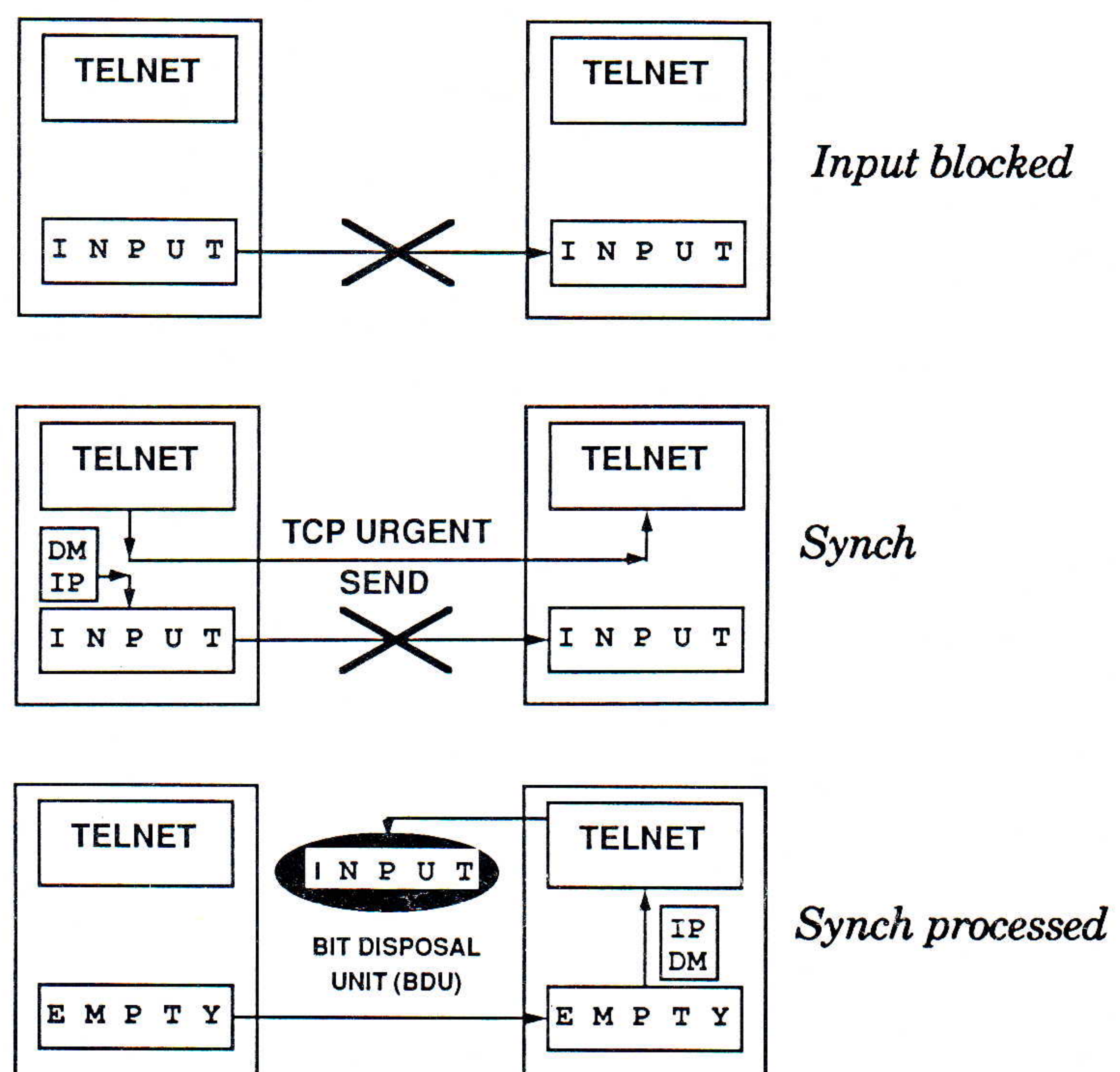


Figure 1: Three stages of synchronization

The Telnet Protocol (*continued*)

An important convention is the combination [IP, SYNCH] to quickly stop a runaway process regardless of any other pending input.

All but IP are optional and can be ignored if they have no local meaning on the host system. These control functions are transmitted as special two-byte sequences (e.g., IAC EL, not as ASCII characters. Do not confuse these with the initiation of these control functions by the user transmitting an ASCII character from the real keyboard, such as hitting ^U which might cause the local Telnet to transmit an Erase Line command.

The NVT printer and keyboard

The width of virtual carriage and length of virtual page are unspecified by default. The 95 USASCII graphics codes (32 through 126) can be displayed on the printer (or screen.) The following of the remaining USASCII codes (0 through 31) are interpreted as:

NUL	No operation
LF	next print line
CR	leftmost margin
BEL	no motion, audible or visible signal
BS	one position left
HT	next (unspecified) horizontal tab stop
VT	next (unspecified) vertical tab stop
FF	top of next page

All other codes cause no effect and are interpreted as no-ops.

The USASCII sequence <CR><LF> is treated specially to mean go to the beginning of the next line (newline) even if that is not the sequence used for this operation on the remote host. In order to send a carriage return directly the two characters <CR><NUL> must be sent. This is true in either data direction.

The NVT keyboard has virtual (imaginary) keys which can produce all 128 USASCII codes. This virtual keyboard can also generate the various codes previously described (SYNCH, AO, IP, EC and EL.) There is a virtual <BREAK> key which is similar to many system's notion of a <BREAK> or <ATTENTION> key interpreted as a data code not equivalent to any USASCII code.

Just because these virtual keys exist it does not mean they will be interpreted on the remote system in any particular way. For example, there may be no meaning on the remote system for an EL (erase line) character, all that is promised is that this code can be transmitted.

Telnet commands

All Telnet commands are sent over the data stream encoded as two or more bytes. The first byte is always the "Interpret As Command" (IAC) character (255) followed by the command byte. To send the value 255 as data it must be sent twice. For example, the EL control code is transmitted as the two byte sequence "IAC (255) EL (248)."

Options processing

So far the NVT is a intentionally primitive model of a terminal. By the use of Telnet option negotiations features can be added to this basic terminal making it arbitrarily sophisticated and specialized to the context of the particular session needs.

There are two possible forms of options negotiations, those which are completed in one command (conceptually, flags to turn on or off) and those requiring further information be passed between the processes (subnegotiations.) Both forms begin with the same basic options negotiation format consisting of a three byte sequence: IAC (255), the request code, and an option code. The requests are one of the following four codes:

WILL	251
WONT	252
DO	253
DONT	254

WILL indicates a desire to begin performing some specified option, (e.g. "I would like to do local echoing of characters.") DO indicates that you desire the other party to perform some specified option (e.g., "Please perform local echoing.")

The negatives are WONT which means I will not perform that option (e.g., "I can't locally echo") and DONT which means the other side should not perform the option (e.g., "you shouldn't locally echo.")

Typically a refusal either means this option does not make sense on this system (or in the current context, possibly affected by other option settings) or the option is entirely unrecognized or unsupported.

Importantly, via these option negotiations one side is free to ask for options that the other side is completely unaware of since the other side only has to turn them around as WONT or DONT option requests to cancel the request. No knowledge of the request's actual semantics is necessary, merely copying back the code into a WONT or DONT request is sufficient.

If an option requires subnegotiation to pass additional information this can be implicitly understood by the fact that the other side has acknowledged that it supports and will perform this option. Such formats for subnegotiations are pre-determined in the RFCs and other protocol documents.

These two rules together allow great latitude in designing and implementing a Telnet program. Local options and subnegotiations can be added which are not expected to be widely understood. A remote Telnet without these options will be able to indicate this within the standard protocol formats and any extended subnegotiations can be passed easily between cooperating Telnets. All that is desired is that some reasonable behavior be exhibited by the augmented Telnet even if its desired options are refused.

Subnegotiations

Subnegotiations are passed by use of the *SB* (250) and *SE* (240) command bytes (each immediately preceded by an IAC.) These bracket the data to be passed so it can be of arbitrary length and of any format special to that particular option's subnegotiation protocol.

Extended options list

The option 255 has been defined as the *Extended Options List* (EXOPL) code [RFC 861]. A Telnet desiring to use EXOPL first sends a normal DO/DONT/WILL/WONT command with EXOPL as the option code.

continued on next page

The Telnet Protocol (*continued*)

Upon positive acknowledgement extended options can be exchanged using the format:

IAC SB EXOPL DO/DONT/WILL/WONT/<option code> IAC SE

This allows for an additional 256 Telnet options code. The primary motivation for defining this at this time was to ensure that the value 255 for EXOPL was reserved. If further subnegotiations are desired for an extended option the format to use is:

IAC SB EXOPL SB <option code> <parameters> SE IAC SE

Option examples

Some simple options with no subnegotiations are:

Binary Transmission (0): Allow sending of all eight bits of subsequent data bytes to remote applications [RFC 856].

Echo (1): WILL means this side will begin echoing data received, DO is a request for the remote side to begin echoing data sent. Either or both can be in effect simultaneously [RFC 857].

Suppress Go Ahead (3): Normally the NVT uses a half duplex protocol which includes sending a special GO AHEAD character when the remote side is ready to receive more input. This option can be turned off on many systems which support full duplex thus saving the overhead and allowing more complete interactive behavior [RFC 858].

Status (5): Willing to exchange Telnet status information [RFC 859].

Logout (18): Forced logout from one end or the other [RFC 727].

End of Record (25): Will use the IAC EOR (239) two byte end of record code as a record delimiter (if not enabled IAC EOR is ignored, this option is symmetric and must be separately negotiated by both sides) [RFC 885].

Some options which use subnegotiations are:

Window Size (31): Request to set or change window size [RFC 1073]. The subnegotiation takes the form:

IAC SB 31 <16-bit width> <16-bit height> IAC SE

Terminal Type (24): Request to set or change the terminal type [RFC 1091]. The subnegotiation uses the format:

IAC SB 24 TERMINAL-TYPE-STRING IAC SE

Where the TERMINAL-TYPE-STRING can be any character string. Both sides need to use the same strings so many terminal type names are officially defined in the latest Assigned Numbers RFC (at the time of this writing [RFC 1010].) Some terminal type names are DEC-VT100, IBM-3278-4 and TELETYPE-33. There are over 150 names currently defined. The maximum length of a name is 40 characters and must start with a letter and end with a letter or digit and be composed of a combination of upper-case letters, digits and/or the hyphen or slash character.

Finally, the most recent Telnet option, 4/1/89, is:

Subliminal-Message Option (257): Negotiate the display of subliminal messages [RFC 1096]. The subnegotiation uses the format:

IAC SB 257 <16-bit value> <16-bit value> <string> IAC SE

The first value is the duration in milliseconds the message is to be displayed. The second value is the frequency with which the message is to be displayed. Finally the string contains the data characters to be displayed. Reportedly this was developed at CMU in an attempt to stop their students and staff from submitting RFCs such as this. For comparison see [RFC 748, 4/1/78], the *Telnet Randomly-Lose Option*.

References

- Miller, B., "Telnet subliminal-message option," RFC 1097.
- Marcy, G.A., "Telnet X display location option," RFC 1096.
- VanBokkelen, J., "Telnet terminal-type option," RFC 1091.
- Hedrick, C., "Telnet remote flow control option," RFC 1080.
- Hedrick, C., "Telnet terminal speed option," RFC 1079.
- Waitzman, D., "Telnet window size option," RFC 1073.
- Levy, S.; Jacobson, T., "Telnet X.3 PAD option," RFC 1053.
- Yasuda, A.; Thompson, T., "Telnet Data Entry Terminal option: DODIIS implementation," RFC 1043.
- Rekhter, J., "Telnet 3270 regime option," RFC 1041.
- Nedved, R., "Telnet terminal location number option," RFC 946.
- Silverman, S., "Output marking Telnet option," RFC 933.
- Anderson, B.A., "TACACS user identification Telnet option," RFC 927.
- Postel, J.B., "Telnet end of record option," RFC 885.
- Postel, J.B.; Reynolds, J.K., "Telnet extended options: List option," RFC 861.
- Postel, J.B.; Reynolds, J.K., "Telnet timing mark option," RFC 860.
- Postel, J.B.; Reynolds, J.K., "Telnet status option," RFC 859.
- Postel, J.B.; Reynolds, J.K., "Telnet suppress go ahead option," RFC 858.
- Postel, J.B.; Reynolds, J.K., "Telnet echo option," RFC 857.
- Postel, J.B.; Reynolds, J.K., "Telnet binary transmission," RFC 856.
- Postel, J.B.; Reynolds, J.K., "Telnet option specifications," RFC 855.
- Postel, J.B.; Reynolds, J.K., "Telnet Protocol specification," RFC 854.
- Postel, J.B., "Remote User Telnet Service," RFC 818.

continued on next page

The Telnet Protocol (*continued*)

Postel, J.B., "Transmission Control Protocol," RFC 793.

Killian, E., "Telnet Send-location option," RFC 779.

Greenberg, B., "Telnet SUPDUP-Output option," RFC 749.

Crispin, M., "Telnet randomly-lose option," RFC 748.

Crispin, M., "Telnet SUPDUP option," RFC 736.

Crocker, D.; Gumpertz, R.H., "Revised Telnet byte macro option," RFC 735.

Day, J.D., "Telnet Data Entry Terminal option," RFC 732 (Updated by RFC 1043).

Day, J.D., "Minor pitfall in the Telnet Protocol," RFC 728.

Crispin, M., "Telnet logout option," RFC 727.

Postel, J.B.; Crocker, D., "Remote Controlled Transmission and Echoing Telnet option," RFC 726.

Internetworking With TCP/IP—Principles, Protocols, and Architecture, by Douglas E. Comer, Prentice-Hall, ISBN 0-13-470154-2

Telnet and telnetd programs for Unix, University of California, Berkeley, available as complete program source from various archive sites. Highly recommended reading.

BARRY SHEIN received his B.Sc. from Cornell University and M.A. from Boston University where he is a Ph.D. candidate in the Department of Computer Science. He is Executive Vice President and co-founder of Software Tool & Die. Previously he worked at Encore Computer Corporation and before that at Boston University as Manager of Special Projects in their Distributed Systems Group for several years. He serves on the Board of Directors of the Sun User Group and has been active in the field of UNIX and networking for over a decade.

Learn more about Telnet! Attend the Telnet session at



The session, S13, is at 10:30am on Thursday, October 5, 1989.

Telnet Alert!

The *Computer Emergency Response Team* (CERT) issued an Internet Security Advisory on Telnet as this issue of *ConneXions* went to press. A hacked version of Telnet which captures passwords has been found at several sites. Administrators who discover any unauthorized activity are urged to contact CERT, via electronic mail at: cert@sei.cmu.edu or the 24 hour hotline: (412) 268-7090.

Components of OSI: CLNP or *A Day in the life of Ivan CLNPacket*

by Rob Hagens, University of Wisconsin

Introduction

An OSI *End System*. Layered. Complicated. This was a place where packets lived a hard and fast life. This is the story of one particular packet. This story is not a unique one. It happens every day. It may already have happened to you.

Ivan, a CLNP packet from birth, worked on the *Connectionless Squad* (CL). His squad leader, Mikhail, had been running the Connectionless network layer for 15 years. Ivan was lucky to be in the CL squad. He had heard all the stories from Ivan about the other guys in the compound—the *Connection-oriented Squad* (CO). The Gosip was that life in the CO squad was brutal. The CO squad leader enforced his reliable, guaranteed delivery service with an iron fist.

Luckily, Mikhail was fair, Ivan diligent. The absence of a delivery guarantee for CLNP packets promoted a laissez-faire atmosphere in the CL squad which permeated all its members. Yes, there was envy from the CO squad, but lack of CO/CL communication meant that the chance of a transfer between them was small.

The Squad Room

Ivan was already bored with Mikhail's briefing. He already knew that he was a full CLNP packet and not a member of either of the two subsets of the CLNP family. The non-segmenting subset, thought Ivan, was only for those packets who were so sure that they would not be segmented that they opted to save 6 bytes by not carrying the segmentation information along. This segmentation information would be used if the packet was split into several, smaller packets by an *Intermediate System*. This process, (*segmentation*) was used if the packet was too large for a transmission link to handle. The separate pieces of the packet would journey toward their destination independently—to be merged together once they reached the destination.

Although Ivan was secretly frightened by the thought of being carved up into many pieces by some unknown, butcher router, he tried to project an outward excitement about the segmentation experience.

Of course, Ivan was very happy he was not a member of the Inactive network layer subset. He had heard stories about these strange relatives with a 1 byte header containing only an identification field. What a strange life it would be—inactive until the end! Better to die while living...

The header

Ivan looked at his header in the hunk of stainless steel on the wall that served as the squad's mirror. The fixed part was complete: his header length set, his lifetime filled with a comfortable value of 30, his type set to normal data, and his checksum (courtesy of Fletcher) computed. The checksum, of course, was only computed on his header. Ivan liked this. He would only worry about his header. The data he carried belonged to a transport connection. He was glad to let the transport connection worry about their data with their own checksum.

continued on next page

A Day in the life of Ivan CLNPacket (*continued*)

Data. *Normal data*. That was Ivan's type of packet. Better to be a normal data packet than a ghastly *Error Report*. No one liked the bearer of bad news. It was a dangerous profession; especially in this compound. It was much better to transport data than the remains of some unknown, discarded packet.

The other parts of Ivan's header were in place. His addressing part was complete with a source and destination NSAP address. His segmentation part initialized and ready to be used if (his mind raced at the thought) segmentation were to occur.

The final section of Ivan's header, the options part, contained a few of the possible CLNP options.

"You will notice, Ivan..." Mikhail's voice brought Ivan's attention back to the front of the room, "...that I have given you several optional parameters. I have added the *Padding Option* to increase your total header length to a multiple of 4 bytes. You also have a *Source Route* and *Record Route* option. You will visit each router in the source route list during your journey through the network. The Record Route option will be used to log the addresses of all the routers you visit. Every time you visit a router, he will write his address in the space I have created."

"Why is the source route so small and the record route so large?" asked Ivan.

Mikhail explained that Ivan's source route was not a complete listing of every router that must be visited. Rather, as a partial source route, it only listed a few of the routers to be visited.

"So I can visit as many routers as I wish, as long as I visit the ones listed in the source route, in the order listed?" interrupted Ivan.

"That's right," Mikhail said. "But keep in mind that your record route option is fixed in size."

"Huh?"

"Your record route option is large enough to contain the address of many routers. However, its size was fixed when you were created. The option can not expand during your journey. If you visit so many routers that your record route option is filled with all the addresses, the recording will have to be terminated."

"Well, I'll be careful." reassured Ivan.

"Indeed. And watch your lifetime. It will decrease as you travel. If it reaches zero, you'll be discarded."

"Don't worry, Mikhail. My lifetime is 30! I have plenty of time."

With that, Ivan scrambled out the door, ready to begin his journey.

"That's what the last packet thought as well," Mikhail whispered.

The Network Swamps

The first few routers that Ivan visited were not very busy and the trip was uneventful. Whenever Ivan arrived at an Intermediate System, that system would write its address into Ivan's record route option, and Ivan's lifetime field would be decremented. With a new header checksum computed, Ivan was immediately forwarded out onto a link towards his destination. In fact, until he reached the long-haul backbone, Ivan didn't even notice the steady tick-tick-ticking of his lifetime field.

Trouble

The moment Ivan appeared on the backbone router's input queue, he felt that something was wrong. The heat was intense. The room noisy. He stood in line trying to gauge the length of the queue. It looked like 6, maybe 7 packets were already waiting. Judging by the size of their lifetime fields, many had been traveling for a long time. Ivan waited impatiently to get closer to the front of the line. Suddenly, as he was about to step up to be routed, 2 other packets jumped in front of him.

"Wait a minute!" Ivan started to shove. The big, blue router looked down.

"Priority Packets, kiddo. If ya don't have a priority option, yer priority is low."

Ivan waited for the priority packets to leave. He stepped up to the router.

"Destination?"

Ivan told him.

"Ok. T'ird link from da left. Ya's been here awhile. Cut da lifetime field by foah."

"Four, FOUR?," Ivan couldn't believe his ears.

"Look fella," the router mumbled, "yer a CLNP packet. Dat means yer lifetime ain't no hop count. It's a time field. And ya been here a while."

Ivan ran to the link. His lifetime was down farther than he thought possible. He knew that if the lifetime field dropped to zero, he would be discarded. Like many packets, he didn't think it could happen to him. He wasn't ready to die. As he jumped down the link, he wondered if he would see those quiet regional networks again...

The next few routers on the backbone were no better. Long delays at each one spelled trouble. One of the routers was so congested that it was dropping packets at random. Ivan was horrified to watch the packet in front of him get discarded, wrapped up in an Error Report, and shipped back towards its source. Other packets at that router were getting their *Congestion Experienced* bits turned on.

When it was Ivan's turn, the router looked at him with distaste.

"You don't have a *Quality of Service Option*."

"I don't?" Ivan half questioned.

"No. You don't."

Ivan did not understand that the congestion experienced bit was part of the Quality of Service Option. Since that option was not required on every packet, it wasn't always available to be set.

"We backbone routers rely on the QOS congestion experienced bit to inform the transport connections that we are congested. What part of the network do you come from, anyway?"

Ivan started to list his record route option. The bored router interrupted,

"Never mind. Begone. Last link on the right."

continued on next page

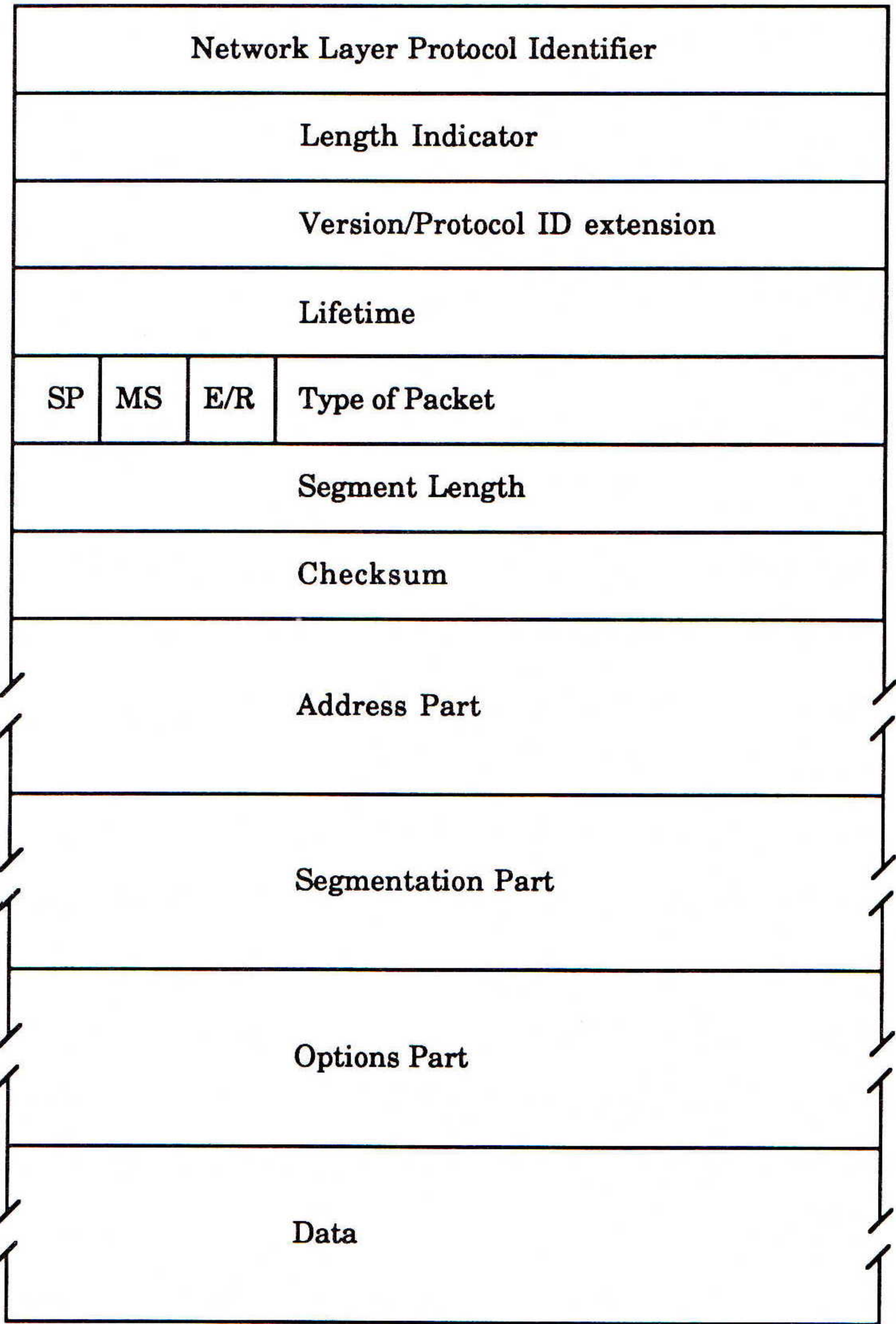
A Day in the life of Ivan CLNPacket (continued)

The final link

Ivan never made it to his destination. His lifetime expired somewhere between the far side of the backbone and his destination subnetwork. His data was discarded. His header not even returned to the compound, for the system that discarded him did not care to send an Error Report.

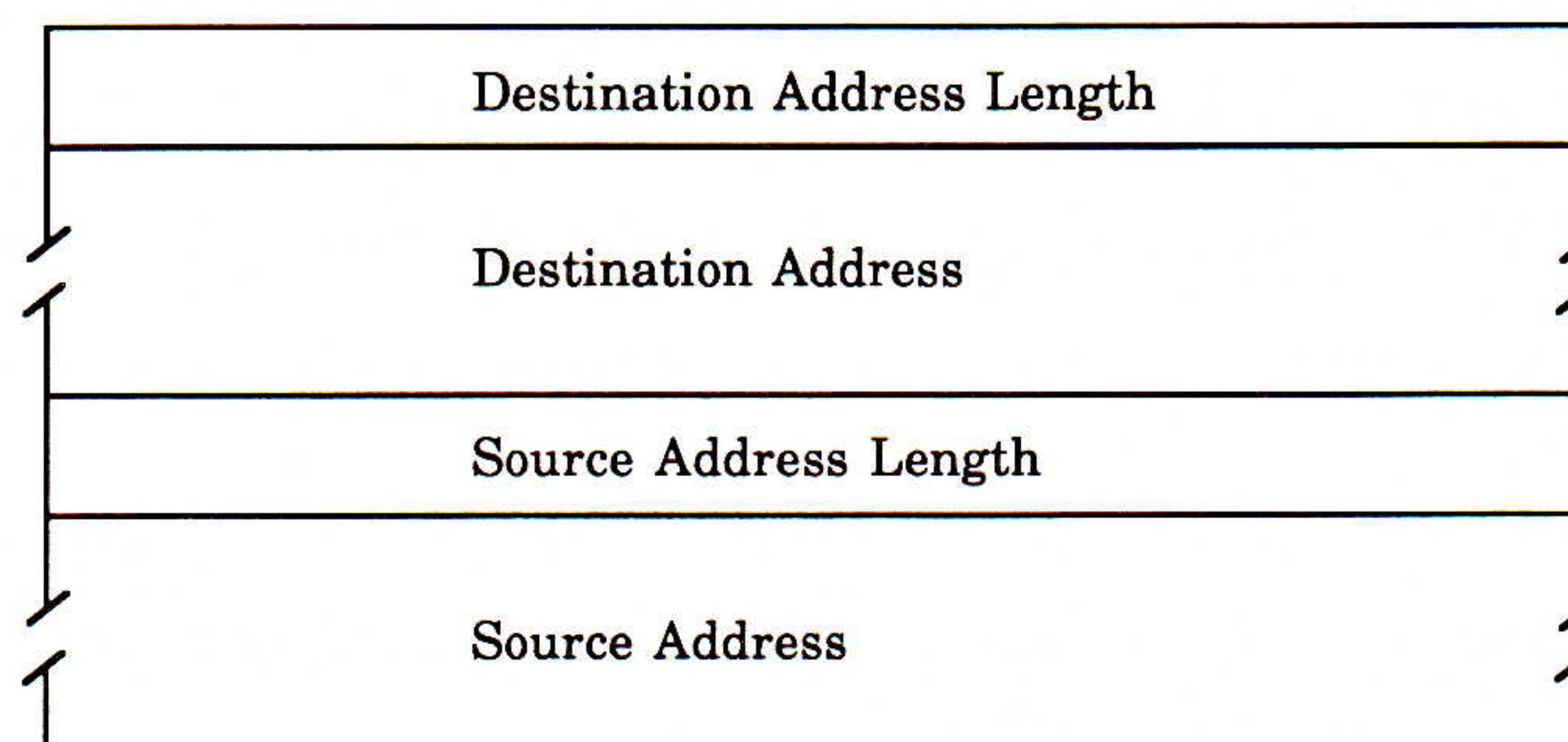
Ivan’s epitaph read:

*Run, don’t plod.
Take it from me.
Behold, behold the links I trod
for ISO 8473.*

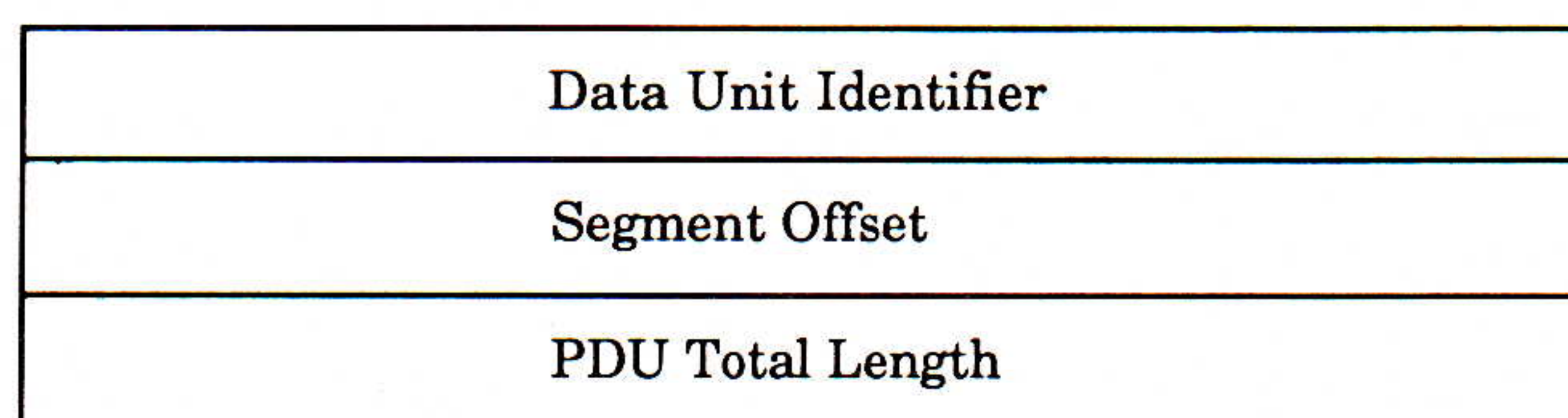


The CLNP packet

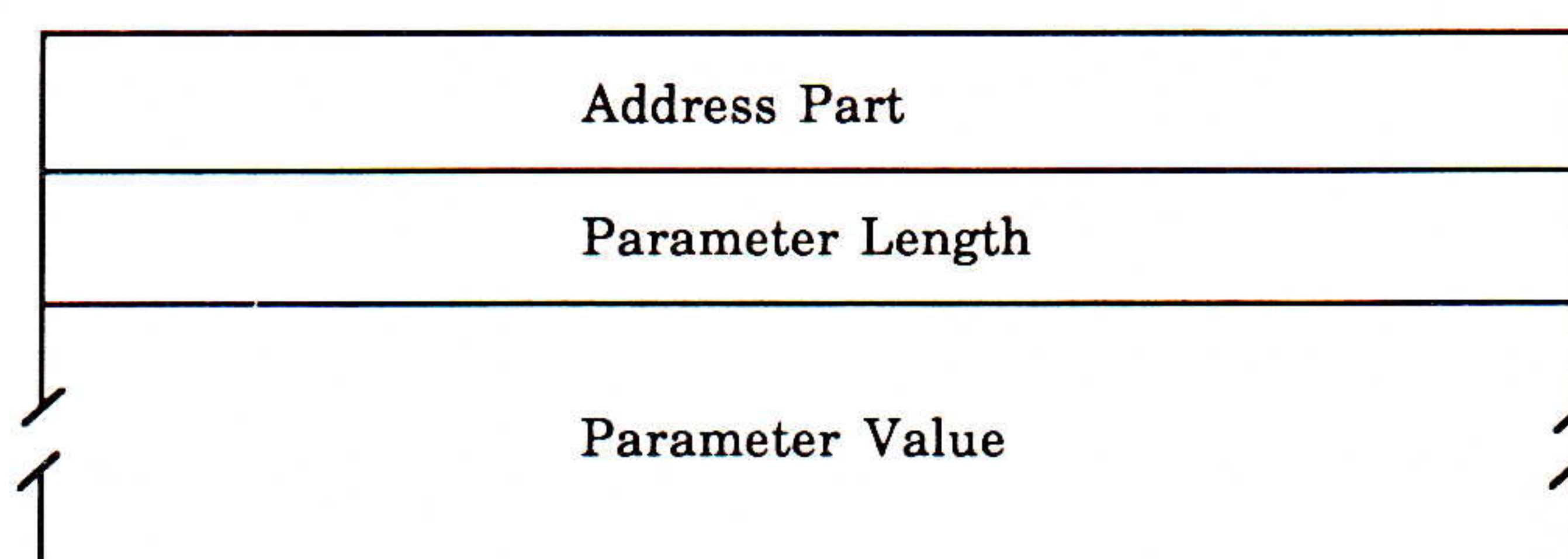
A CLNP packet consists of a header followed by data. The header can be broken into a 9 byte fixed part, followed by the *Address Part*, the *Segmentation Part*, and the *Options Part*. The maximum size of the header is 254 bytes. Both the Normal Data and Error Report use the same format. These packets are distinguished by their *type* fields.

**Address Part**

The *Address Part* contains the destination and source NSAP addresses. ISO 8473 does not define the address format. The format used by CLNP is defined in addendum 2 of the Network Service Definition, ISO 8348. The maximum length of an NSAP address is 20 bytes; the maximum length of the address part is 42 bytes.

**Segmentation Part**

The *Segmentation Part* is used to identify information about a particular piece of a segmented CLNP packet. Presence of the Segmentation Part does not imply that the packet *must* be segmented—only that it *may* be segmented. If the non-segmenting subset of the protocol is employed, the Segmentation Part is not included in the PDU. Absence of the Segmentation Part is indicated by a zero in the SP flag.

**Optional Parameters**

Optional parameters are specified as a series of code, length, value tuples. The following optional parameters are defined for CLNP: padding, security, source routing, recording of route, quality of service, and priority.

ROBERT A. HAGENS received an M.S. in Computer Science from the University of Wisconsin–Madison in 1984. Hagens is currently an Associate Researcher in Network Communications at the University of Wisconsin–Madison. His current work interests are in user interface design, especially when related to electronic mail. He is also the co-chair of the IETF OSI working group. From 1985 to 1988 he implemented the ISO Session and Network layer protocols and provided UNIX kernel systems support for the Wisconsin OSI project. In 1984, Hagens worked on the WISNET project (DoD protocol suite for IBM VM/SP systems) including implementation and performance improvements to FTP, UDP, and TCP/IP.

CONNEXIONS

480 San Antonio Road
Suite 100
Mountain View, CA 94040
415-941-3399
FAX: 415-949-1779



FIRST CLASS MAIL
U.S. POSTAGE
PAID
SAN JOSE, CA
PERMIT NO. 1

CONNEXIONS

PUBLISHER Daniel C. Lynch

EDITOR Ole J. Jacobsen

EDITORIAL ADVISORY BOARD Dr. Vinton G. Cerf, Vice President, National Research Initiatives.

Dr. David D. Clark, The Internet Architect, Massachusetts Institute of Technology.

Dr. David L. Mills, NSFnet Technical Advisor; Professor, University of Delaware.

Dr. Jonathan B. Postel, Assistant Internet Architect, Internet Activities Board; Division Director, University of Southern California Information Sciences Institute.

Subscribe to CONNEXIONS

U.S./Canada \$125. for 12 issues/year \$225. for 24 issues/two years \$300. for 36 issues/three years

International \$ 50. additional per year (Please apply to all of the above.)

Name _____ Title _____

Company _____

Address _____

City _____ State _____ Zip _____

Country _____ Telephone () _____

☐ Check enclosed (in U.S. dollars made payable to CONNEXIONS).

☐ Charge my ☐ Visa ☐ MasterCard ☐ Am Ex Card # _____ Exp. Date _____

Signature _____

Please return this application with payment to:

CONNEXIONS

480 San Antonio Road Suite 100
Mountain View, CA 94040

415-941-3399 FAX: 415-949-1779

Back issues available upon request \$15./each
Volume discounts available upon request

CONNEXIONS